INCREASED GROUND VIBRATION MEASUREMENT SPEED FOR LANDMINE DETECTION
FINAL PROGRESS REPORT

JULY 9, 2009

SUBMITTED TO:

ARMY RESEARCH OFFICE
ATTN: AMSRD-ARL-RO-SS-SI (TR)
P.O. BOX 12211
RESEARCH TRIANGLE PARK, NC 27709-2211

BY:

JAMES M. SABATIER
NATIONAL CENTER FOR PHYSICAL ACOUSTICS (NCPA)
UNIVERSITY OF MISSISSIPPI
UNIVERSITY, MISSISSIPPI 38677

PERIOD: 18 APRIL 2007 – 17 APRIL 2009
CONTRACT NUMBER: W911NF-07-C-0048
SECURITY CLASSIFICATION: UNCLASSIFIED

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **09 JUL 2009**	2. REPORT TYPE	3. DATES COVERED **00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE **Increased Ground Vibration Measurement Speed for Landmine Detection**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **National Center for Physical Acoustics (NCPA),University of Mississippi,University,MS,38677**	8. PERFORMING ORGANIZATION REPORT NUMBER **; 52580.1**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **U.S. Army Research Office, P.O. Box 12211, Research Triangle Park, NC, 27709-2211**	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S) **52580.1**

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**94**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Increased Ground Vibration Measurement Speed for Landmine Detection

Abstract

The objective of this program, to demonstrate the feasibility of time-division multiplexing for rapid ground vibration measurements, has been achieved. Time-division multiplexing is a technique in which a multiple-beam laser Doppler vibrometer (MBLDV) passes over the ground while each beam measures a portion of the ground vibration. The signal from each beam is combined to determine the vibration pattern of that section of ground. In this way, the ground vibration measurement speed can be increased in a manner directly proportional to the number of beams in the down-track direction.

The University of Mississippi (UM), working with MetroLaser, Inc. and Planning Systems Incorporated, adjusted a MBLDV to accommodate new broadband photodetectors, developed data acquisition and demodulation software, tested the overall measurement system using simulated motion of a vibrating target, built a test track, acquired data with a moving sensor, and tested a basic demultiplexing algorithm. Data was taken and processed to a velocity profile proving the concept of time division multiplexing with a MBLDV. The results indicate promise for this technique and a follow-on grant has been awarded to UM to expand the capabilities of the hardware and software to further this research in more realistic environments.

Table of Contents

List of Appendixes

Statement of the Problem Studied

UM studied the problems associated with increasing the scanning speed of a moving laser Doppler vibrometer (LDV), which include overcoming spatial and frequency resolution-imposed limits of scanning speed, speckle noise, Doppler offset due to vehicle speed, and platform motion.

The first and biggest problem is the minimum dwell time required over a ground segment to achieve the necessary frequency resolution. This defines the forward speed to be the product of the spatial resolution and the frequency resolution as shown in the expression

$$S_B = s * \Delta f \qquad (1)$$

where S_B is the speed of the beam, s is the spatial resolution and Δf is the frequency resolution. From Equation (1), the beam must move over the target area of length s during the time T, where $T = 1/ \Delta f$. Therefore, if the spatial resolution is to be no larger than seven centimeters and the frequency resolution to be no less than five hertz, then the maximum forward speed will be 35 cm/s.

The next problem is the increased noise floor due to speckle noise as the laser beam moves over a diffuse reflective surface. Speckle noise is a lesser problem for LDV systems that are stationary, and previous scanning systems tested in the field have been situated on heavy vehicles to minimize platform motion. However, once the system becomes mobile, speckle noise can become a significant part of the measured signal.

Doppler offset has not been a factor of previous scanning systems tested in the field since the systems were stationary. Once the LDV is moving, then the Doppler offset is defined by

$$F_d = \frac{2V}{\lambda}\cos\theta \qquad (2)$$

where F_d is the frequency offset due to Doppler shift, V is the velocity of the vehicle, λ is the wavelength of the laser light and θ is the angle between the direction of the laser beam and the direction of the velocity vector of the vehicle. The Doppler offset due to vehicle motion requires that the laser and photodiode electronics have a wide enough bandwidth to handle the offsets produced while the vehicle is moving. With a desired system speed of 6 m/s and a beam angle relative to the velocity vector of 79 degrees front and back, the maximum Fd will be close to +/- 4.3 MHz. Therefore, a LDV with a 10 MHz carrier frequency will experience a Doppler-adjusted carrier frequency in the 5.7 to 14.3 MHz range.

Platform motion will cause issues in performing small vibrational ground measurements. This problem occurs when small surface vibrations ride on top of the large, low-frequency platform-induced vibrations from the interaction of the vehicle suspension to surface irregularities and platform accelerations. Therefore, the LDV system must have sufficient dynamic range to resolve these two disparate vibration sources.

Approach

The approach taken by UM to address each of the problem areas is outlined below.

Time division multiplexing to increase platform speed. Time division multiplexing uses a LDV with N multiple beams in the line of travel. With time division multiplexing, the speed of motion of the MBV can be increased N times without worsening the spatial and frequency resolutions. The distance between beams is equal to the required spatial resolution. The beams move in the down-track direction and measure the vibration of each space segment sequentially. The duration of a time signal produced by each beam during its motion over each space segment **s** is N times shorter than the required time. Next, the time domain signals from each beam passing over a certain space segment **s** are combined together to obtain a time segment of required duration corresponding to that space segment, just as for one beam. Then the spectrum of vibration is calculated for each combined time segment. The speed of beams is N times higher than the speed of the single beam LDV. This is illustrated in Figure 1.

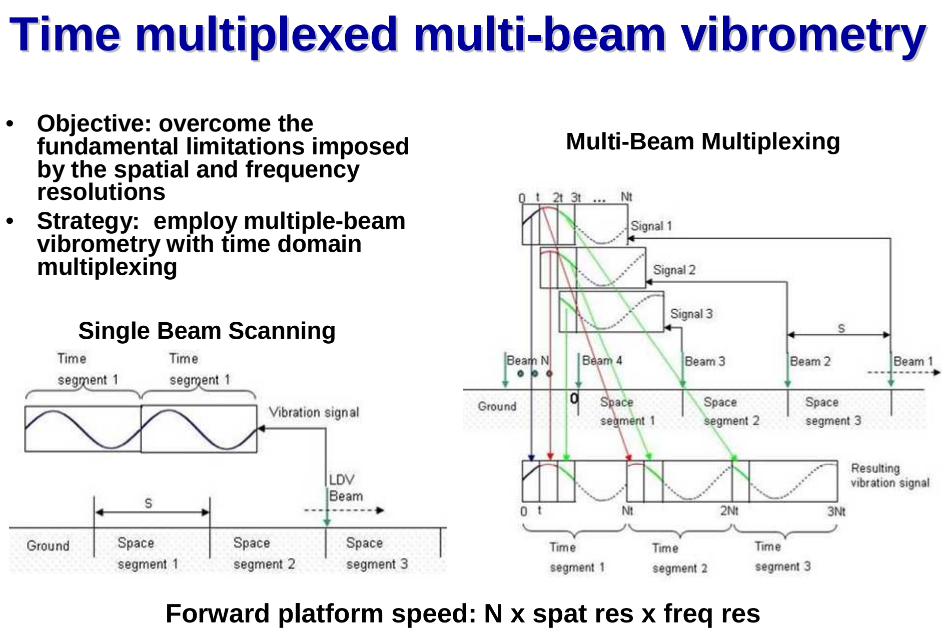


Figure 1. Concept of time multiplexed multi-beam vibrometry.

Speckle noise mitigation. Initial tests indicate that the speckle noise of a moving LDV beam rapidly reaches an upper limit and is not linearly related to the speed of the moving beam. Several different approaches to signal processing methods for speckle noise reduction have been developed from previous research efforts. These approaches have been leveraged and refined for the current application. These include a nonlinear "spike-removal" filter, a method based on carrier signal amplitude, and a method based on wavelet transforms.

Doppler offset due to vehicle speed. Each of the laser beams has an offset from the base carrier frequency. The amount of offset is determined by the speed of the vehicle and the angle of the beams. The beams that are closer to normal incidence (ones in the middle) have the smallest

offset, while the beams at with the largest angles (ones at the ends) have the largest offset. The beams fan out from the LDV at +/- 11 degrees when the system is mounted so that the center beams (beams 7 and 8) are closest to being vertical. In this arrangement, the initial calculations show that at the target speed of 6 m/s, the offset can be as much as +/- 4.3 MHz. A tracking mechanism has been implemented to track the carrier frequency to optimize the in-phase and quadrature (I and Q) clock frequency for each channel.

Technical Significance and Army Relevance

The University of Mississippi has developed acoustic/seismic methods for landmine detection that rely on laser Doppler vibrometry. Energy from loudspeakers or shakers enters the ground and causes landmines to resonate at their natural frequencies. This causes increased vibration on the ground surface that is localized over the mine. The vibration of the ground surface is measured over an area using a laser Doppler vibrometer. The scanning speed is limited by the vibrometer and is currently about 20 seconds per square meter. Time division multiplexing allows dramatically higher scanning speeds with potential rates of advance up to 20 kilometers per hour.

There are numerous benefits to the Army for high scanning rates of surface vibrations. Current research at the University of Mississippi includes detection of vehicles obscured by forest foliage, passive acoustic methods for tunnel detection, passive acoustic methods for detecting anomalies in levees, and detection of improvised explosive devices.

The development of the test apparatus and software for this effort is essentially completed. Initial tests are continuing.

Summary of Important Results

The University of Mississippi has demonstrated a time-division multiplexing capability using a modified multiple-beam laser Doppler vibrometer mounted on a moving vehicle.

- The bandwidth of a 16-beam multiple beam vibrometer has been increased to accommodate increased vibration detection speeds. The vehicle carrying the landmine detection apparatus is expected to travel at speeds up to 20 kilometers per hour. The angle of the beams with respect to the vertical varies up to approximately +/- 11 degrees. This requires a bandwidth for the vibrometer photodetector of greater than 4.4 MHz. MetroLaser, Inc., a subcontractor to UM, has performed the necessary modifications to achieve this bandwidth and adjustments have been made to the multiple-beam laser Doppler vibrometer to accommodate these changes.
- Collection of this quantity of data requires data acquisition cards with greater capability than previously existed for the multiple beam vibrometer system. The cards have been delivered and installed.
- Data acquisition software has been developed and implemented. The demodulation and signal processing software has been developed and tested against simulated FM signals.
- The overall data collection system composed of the multiple-beam LDV, data acquisition system, and developed software mounted on an electric vehicle was initially tested using

simulated motion of a vibrating target. This was achieved by moving a shaker under stationary data collection system.

- Test tracks with the target surfaces treated with retroreflective glass beads to maximize the intensity of the back scattered light have been designed and built. Data has been taken using the test track. This track is shown in Figure 2.

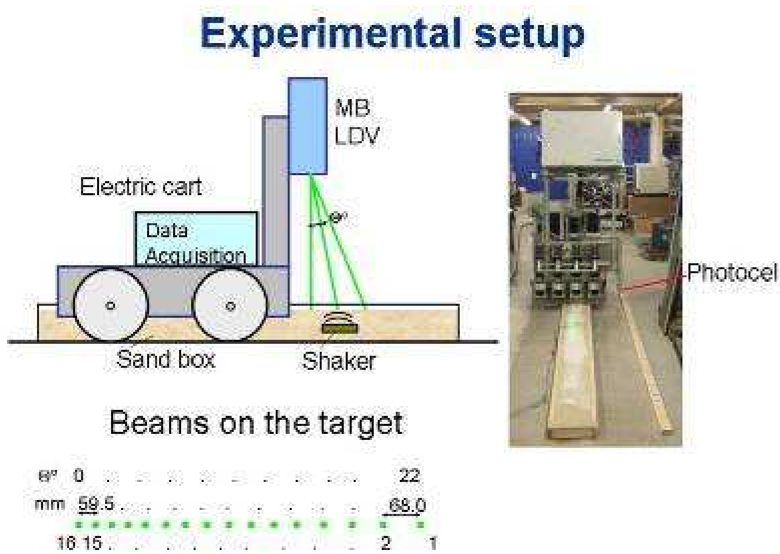


Figure 2. Experimental setup showing the data collection system and a test track with increased reflectivity.

- Signal processing algorithms for speckle noise mitigation have been developed and used in demodulating the data.
- A basic demultiplexing algorithm has been developed and tested using the data taken from the prepared-surface test track. The results are shown in Figure 3.

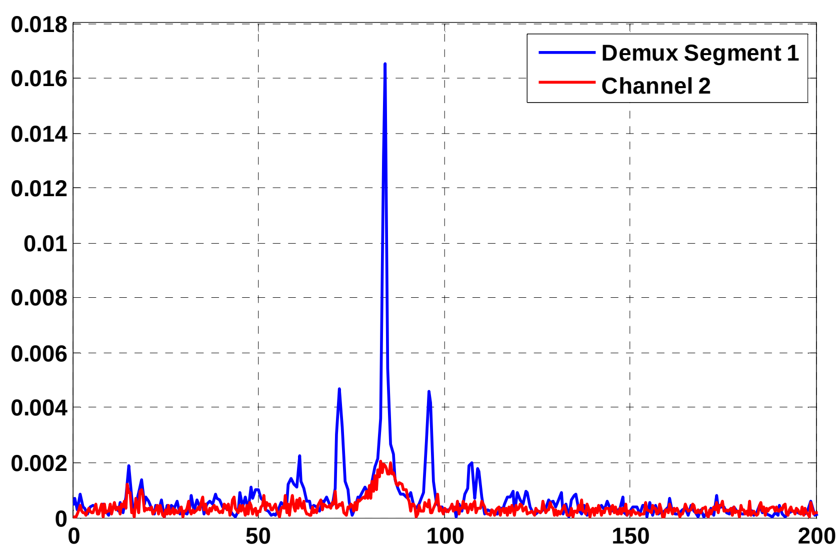


Figure 3. Comparison of demultiplexed data vs. data from a single channel.

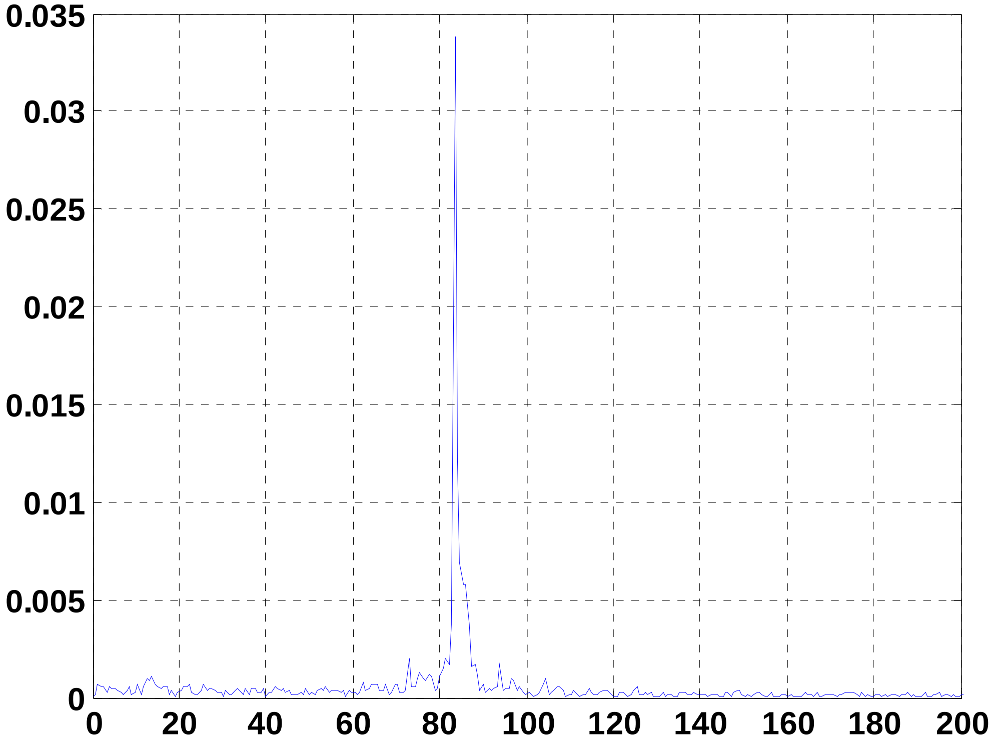


Figure 4. Demultiplexed data based on selecting the segment times using a correlation method.

- Carrier tracking has been implemented to allow the carrier mixing signal to be optimized for each beam every 0.06 seconds. Within this time segment, the carrier frequency is estimated for each beam. This accounts for the constant frequency offset due to the velocity of the vehicle and the angle that the laser beam makes with the ground. This relationship is shown in Equation 2 above.
- Measurement of the system noise floor showed that processing the signals using the full bandwidth of the photodiode electronics is not feasible. Therefore, the carrier tracking algorithm became more important to center a relatively narrow (120 KHz) band pass filter on the carrier of each beam to lower the system noise.

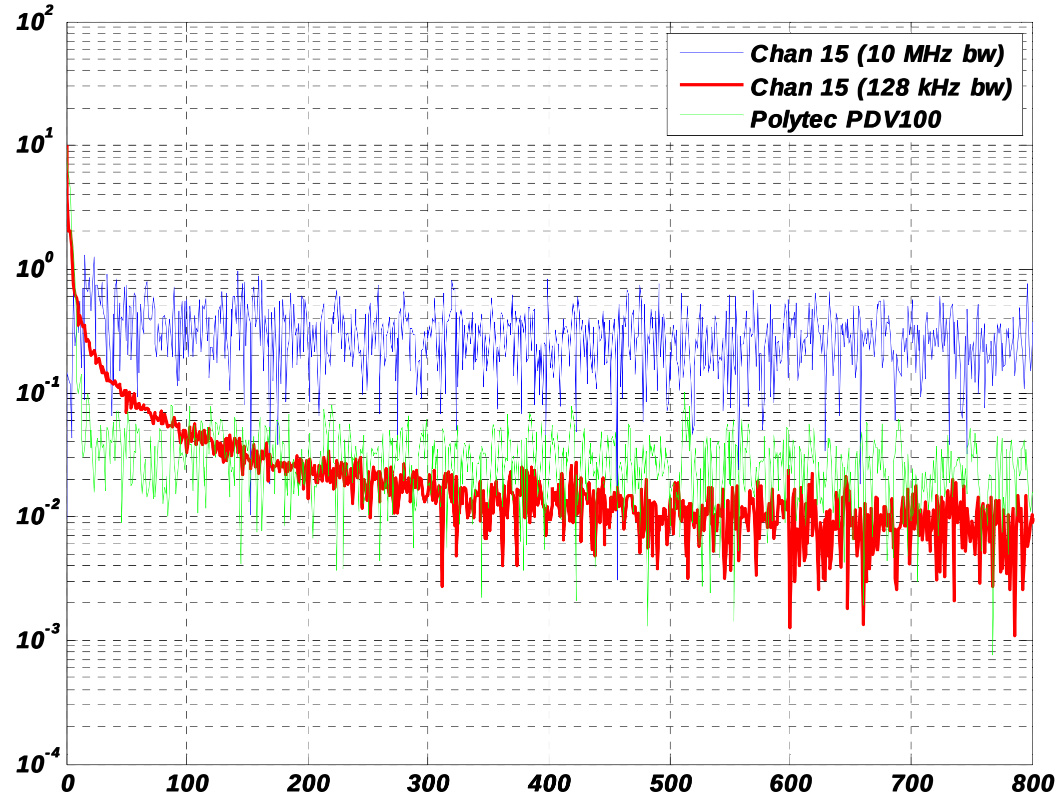


Figure 5. Noise floor comparison based on processing bandwidth.

Technical Discussion

LDV Development

As mentioned in the Approach section and in the Summary of Important Results, the laser and associated photo detector electronics of the multi-beam LDV were modified. The main focus of the modifications was to achieve the increased bandwidth needed to address the Doppler offset, as well as the low-frequency platform motion.

The LDV optics were modified to allow for an increase in the intermediate frequency (IF) from 100 kHz to 10 MHz along with an increase in bandwidth from 40 kHz to +/- 5 MHz. Refer to the LDV Description section for a more detailed explanation of the optical layout. The acousto-optical modulators were changed to obtain the 10 MHz IF. The original acousto-optical modulators were 80 MHz and 80.1 MHz and the new ones are 80 MHz and 70 MHz. Therefore, when the reference and objective beams mix on the photo diode, there is a 10 MHz difference frequency that is detected, filtered and amplified.

The electronics containing the photo diodes were modified to allow for the increase in bandwidth. MetroLaser was responsible for modifying the photo detector to boost both the gain and bandwidth.

LDV Description

The system uses a single mode, solid-state, continuous-wave frequency-doubled (green) Nd:YAG laser with a wavelength of 532 nm and an output power of 200mW. The optical configuration is based on a Mach-Zehnder interferometer with polarization separation of the beams. This layout is shown schematically in Figure 6. The polarizing beam splitting (PBS1) cube is used to divide the linearly-polarized laser beam into an object and a reference beam. The s-polarized component of laser light is reflected by the beam splitter PBS1 to form an object beam, and the p-polarized component of the laser light passes through PBS1 and forms a reference beam. The relative ratio between the object and the reference beams can be adjusted with the half-wave plate HWP1 after the laser. The object beam passes through the half-wave plate HWP2, which changes its polarization from s to p. Then, the object beam is directed to a Diffractive Optical Element (DOE1), which divides it into 16 beams over an angle of 22° (for simplicity, only 3 beams are shown in the figure). The beams pass through the polarizing beam splitter (PBS2) and a quarter-wave plate (QWP) and are directed onto the target by using a fold mirror (M6) and a rotating mirror (M7). The quarter-wave plate changes the polarization of the object beam from linear to circular. The 16 beams illuminate a length of 1 meter, with 6.7 mm separation between the beams. The scanning mirror allows the 16 beams to move in the transverse direction (although this mirror is not moved in this application). The light scattered from the target passes back through the quarter-wave plate, which now changes its polarization from circular to linear s-polarization, leading to a reflection at the polarizing beam splitter PBS2. A collection lens (L1) then collimates the light from each of the 16 object beams. However, since the 16 object beams travel through the center of the collection lens, their path is unaltered and they retain the same angle originally imposed by the DOE.

The reference beam is frequency shifted by means of two Acousto-Optic Modulators (AOMs). The two AOMS operate at 80.0 MHz and 70.0 MHz respectively to produce a net frequency shift of 10 MHz. This frequency shift is obtained by using the +1 order diffracted beam from AOM1 as the input beam for AOM2; the –1 order diffracted beam of AOM2 has a 10 MHz frequency shift relative to the initial frequency of the laser radiation. The reference beam then passes through the half-wave plate HWP3 that changes its polarization from p- to s- and is directed to another Diffractive Optical Element (DOE2). The DOE2 is identical to the DOE1 and divides the reference beam into 16 beams with the same angular separation as the object beam. The 16 object beams and the 16 reference beams are spatially overlapped at the beam combiner (BC), producing 16 heterodyne signals with a carrier frequency of 10 MHz. A final lens (L2), placed one focal distance away from the collection lens, makes the 16 object-reference beam pairs parallel and focuses them onto 16 individual fiber-coupled pin photodiodes.

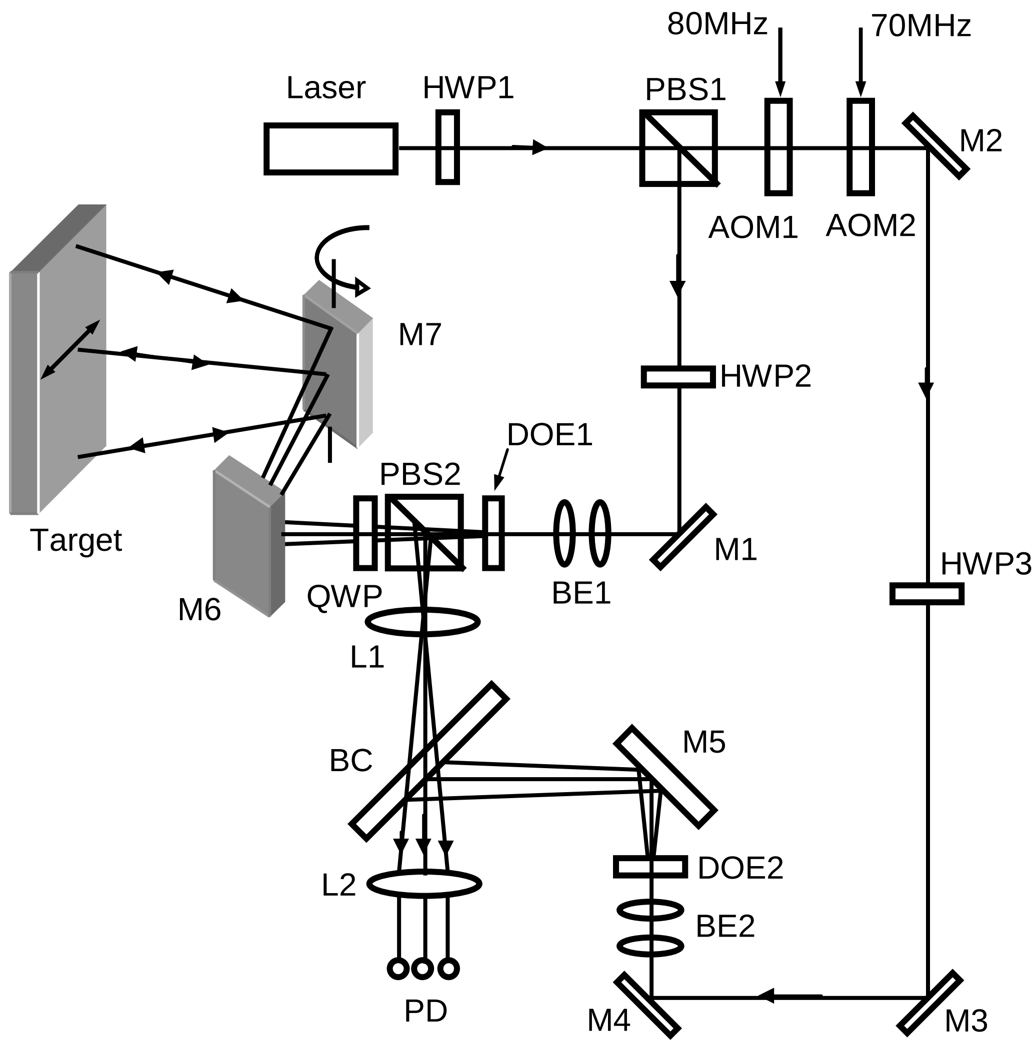


Figure 6. Schematic layout of MB-LDV system

Data Acquisition Development
Hardware

The data acquisition system consists of eight two-channel Signatec PDA14 analog-to-digital (A/D) converter cards and one National Instruments 6120 four-channel data acquisition card (DAQ).

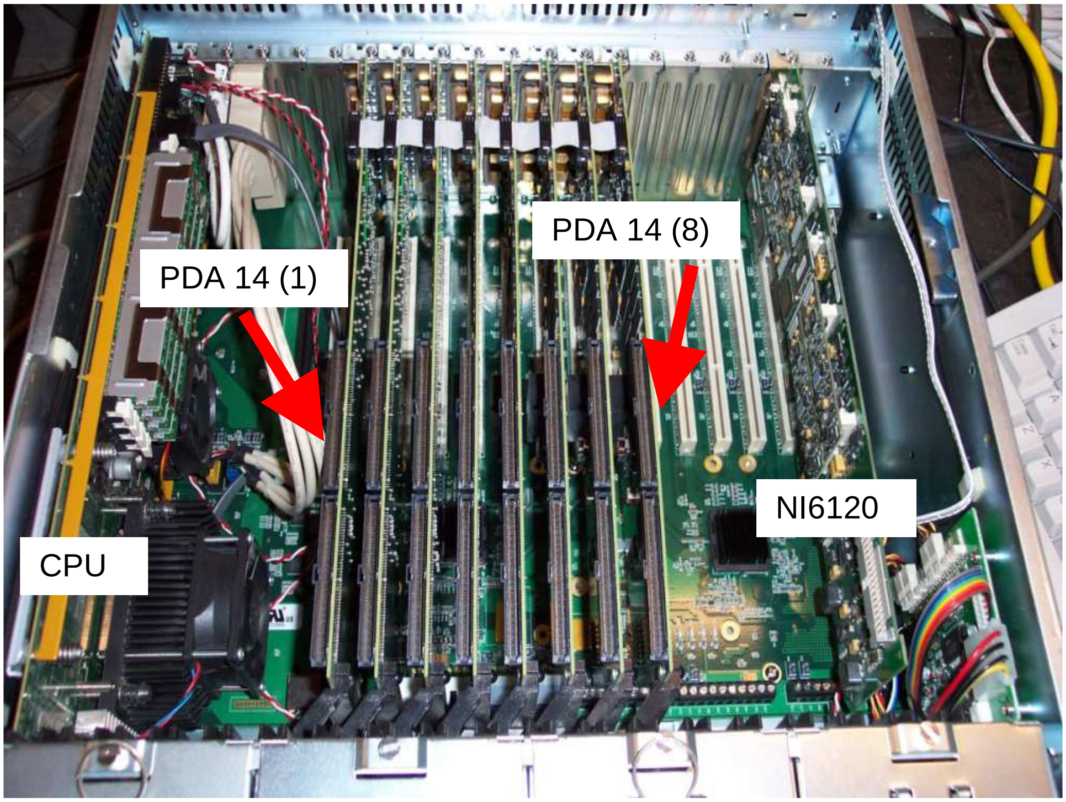


Figure 7. Board placement in CPU Chassis

The Signatec A/D cards are set up so that one card is the master and provides timing and trigger information to the remaining 7 cards. This provides synchronous sampling of all LDV data. The standard configuration that Signatec normally supports is one master card and three slave cards. For the UM application, Signatec provided the ability to control 7 boards by making slight modifications to their board's firmware: the additional cards can be controlled by one master card. Therefore, the cards have to be installed in a defined order as shown in Table 1. Table 1 also shows the master/slave configuration information associated with each card. Figure 7 shows the computer with the top removed to show the installed boards.

The Signatec PDA 14 A/D cards are capable of a maximum sample rate of 100 mega samples per second (MSPS) with a bit depth of 14 bits. The A/D cards have 128 mega-samples per channel of on board memory. With the normal sample rate of 50 MSPS, this provides temporary storage for the data during collection since the collective data rate from all 16 cards (approximately 4,096 million bytes per second) would swamp the PCI bus. Therefore, the data is stored on board the A/D card until the data collection is finished. The limit on how much data can be acquired using this particular system means that the maximum length of a scan can be no more than 2.54 seconds, which places a lower limit on the speed the vehicle.

Board #	Serial #	Function	Master / Slave Assignment
1	404793	Slave	4
2	404904	Slave	4
3	404893	Slave	4
4	404860	Master	1
5	404782	Slave	2
6	404848	Slave	2
7	404804	Slave	3
8	404859	Slave	3

Table 1. Board placement in CPU Chassis

The timing and trigger are provided by a photocell scanning a track with retro-reflective tape. Figure 8 shows this arrangement. The tape is placed at 10 cm intervals along a wooden plank. The photocell provides a positive voltage as it passes over a tape location and goes to nearly zero when not over the retro-reflective tape. The leading edge and falling edge of this voltage define the beginning and end of the tape. This pulse is used to calculate the average speed as the vehicle moves along the track. Since the first tape position provides the trigger signal, this allows the initial trigger point to be adjusted relative to the where the first LDV beam and the target are located.

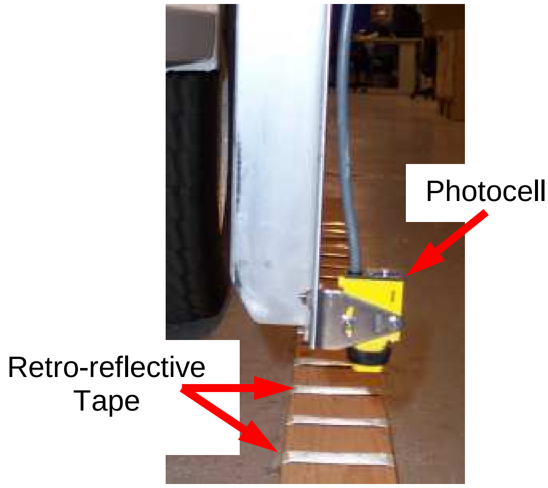


Figure 8. Photocell for trigger and speed measurement

Software
Acquisition User's Guide

LabVIEW from National Instruments was used to perform the basic data acquisition. This data acquisition and processing software takes a graphical approach to designing a user interface as well as designing and setting up the acquisition process.

Figure 9 shows the graphical user interface (GUI) for the acquisition front panel. The main acquisition set up is done in three areas: "Board Configuration", "Trigger Level", and "Speed-Triggering".

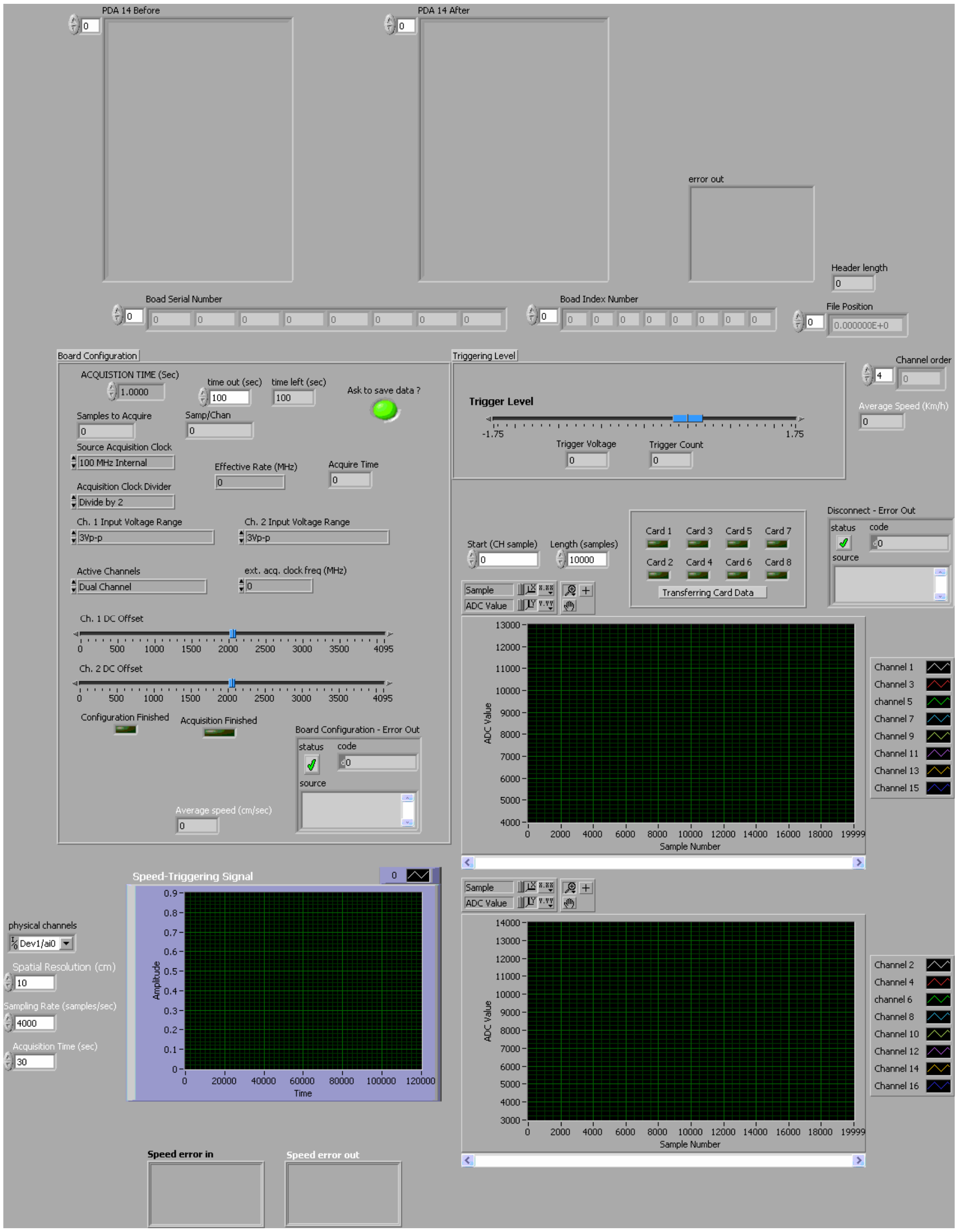


Figure 9. Data acquisition GUI

When the application (DAQ_TDM_spd_8bd.vi) is launched, the part of the GUI that is shown in the window is reduced from what is shown in Figure 8. The user does not see the area above the "Board Configuration/Trigger Level" section at the top, and the error panels at the bottom are cut off as well.

The “Board Configuration” area provides the user with the controls necessary to set up the Signatec boards for the basic data acquisition process. One can set the number of seconds to acquire data (this entry will be checked at run time to ensure that the time entered does not exceed the on-board memory of the A/D card based on the chosen sample rate), acquisition clock (internal or external, and divider), input voltage range, the number of channels per board (one or two), and the amount of offset necessary to zero the cards (this is not a critical adjustment; it needs to be close, but it does not have to be exact). There is also a control entitled, “Ask to save data?” that, if active, will bring up a GUI after the data has been acquired to allow the user to save the data to a file on the computer’s disk storage system.

The “Trigger Level” sets the input voltage level required at the trigger input in order to start acquisition. This level can be set between -1.75 volts and + 1.75 volts. Once the application is started, the slider is read, and the trigger voltage is displayed along with the equivalent count value.

The “Speed-Triggering” section sets the acquisition parameters for the NI-6120 card, which is responsible for acquiring the timing signals from the photocell. The parameters to be set include the channel number, the spatial resolution (how far apart the retro-reflective tape is spaced), the sampling rate, and the number of seconds to acquire data.

The NI-6120 DAQ card and the Signatec A/D card run independently of each other; therefore, the acquisition time specified for the “Board Configuration: area and the “Speed-Triggering” area are different. The acquisition time for the DAQ card acts like a default time out, meaning that, once the application is started, the user has the amount of time specified in the acquisition time control to complete the run. This acquisition is not triggered, whereas the acquisition time for the Signatec A/D in the Board Configuration area, actually sets the total number of samples to acquire (based on the sample rate) after the trigger.

The other areas on the front panel GUI provide information to the user. The areas labeled “PDA 14 Before” and “PDA 14 After” show various state and configuration information about the Signatec PDA 14 A/D card before and after the card’s configuration is set. This allows the user to verify that each card is being set correctly by incrementing the control in the upper left section of each area. The “Board Serial Number” and “Board Index Number” show the number of boards that the software is set to configure based on the number of entries in each array. As mentioned before, the boards are installed in a specific order. The manufacturer’s configuration requirements also state that the master board has to be the last board configured. These arrays allow the software to configure the boards in the correct order. These are mainly informational to aid in troubleshooting.

The display section in the “Speed-Triggering” section will display the data taken by the NI-6120 DAQ card at the end of the specified acquisition time. The first pulse shown is one used as the trigger signal to the Signatec PDA 16 cards and signifies the start of acquisition.

After data acquisition, the software will use the data collected from the photocell to calculate an average speed of the vehicle during the data acquisition phase. This number is displayed at the bottom of the “Board Configuration” area.

The “Transferring Card Data” section shows the progression as the system reads the data from each card to store the data to disk. Once all of the cards have been read, the data is shown on the two displays immediately below the indicators. Odd channels are shown on the top display and even channels are shown on the bottom display. The two controls next to the “Transferring Card Data” section allow the user to select where in the data stream to start selecting data for display and how many samples to display. The number of displayed samples should be kept to approximately 10,000. The actual limit is based on the memory available to LabVIEW. If the number is too large, the application will stop with an “out of memory” error.

There are several error reporting blocks to provide feedback should something go wrong during the configuration, acquisition, or storage phase of collecting data. In addition to the “Transferring Card Data”, status indicator, the “Configuration Finished” and “Acquisition Finished” also show program progress.

The values shown in Figure 9, along with the slider positions, provide the basic system setup for taking data. Three controls may need to be adjusted before taking data: the “Acquisition Time”, “Acquisition Clock Divider”, or the “Input Voltage Range”.

The “Acquisition Time” allows the user to set the number of seconds to acquire data after the trigger. This value is used, along with the “Acquisition Clock Divider”, to calculate the number of samples that each card is to acquire. This value can be set over a range from less than a millisecond to several tens of seconds. The software will check to see if the value entered is too large and adjust it to the maximum-allowed value.

The “Acquisition Clock Divider” allows the user to set the sample rate clock for the A/D with divider values ranging from 1 to 1024. This value, along with the “Acquisition Time”, defines the total number of samples to acquire.

Program Flow

The use of a block diagram GUI to implement the software in LabVIEW does not easily lend itself to external documentation. The screen shots of large complex tasks will not allow the reader to follow the program flow easily, because they will be disjointed. Therefore, the reader is encouraged to bring up the block diagram of the program on a computer and use it as a visual reference. In the course of the following discussion, LabVIEW functions, programs and, subprograms will be referred to as VIs. It should be noted that the following is not intended to be a tutorial on LabVIEW, and it is assumed that the reader has some familiarity with LabVIEW and programming in general.

The initialization section of the wiring block diagram is presented in Figures 10 and 11. Figure 9 shows the user interface controls and front panel indicators that are linked to the wiring diagram in Figures 10 and 11. The data is read from these controls when the program execution is started. Once the program reads the initial configuration data, the time of acquisition input is checked to verify that this time does not exceed the total number of samples that can be stored in the memory of the Signatec A/D boards. There are also a number of constant arrays that tell the

software which Signatec board is a master and which are slaves, along with the specific order of the boards. This information is read and then used to make sure that the cards are configured in the correct order, as the master card has to be configured last.

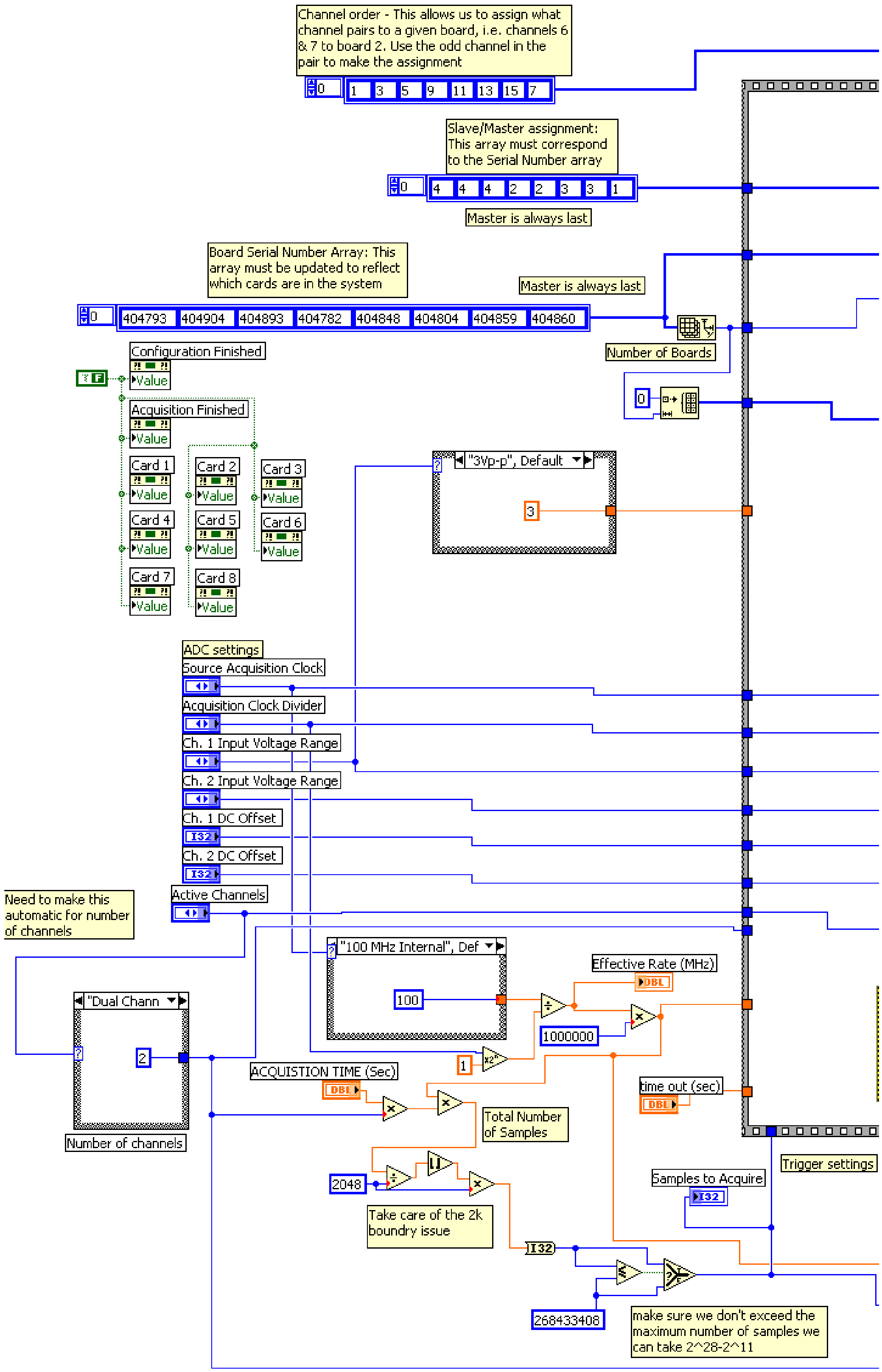


Figure 10. GUI initialization wiring diagram (part 1)

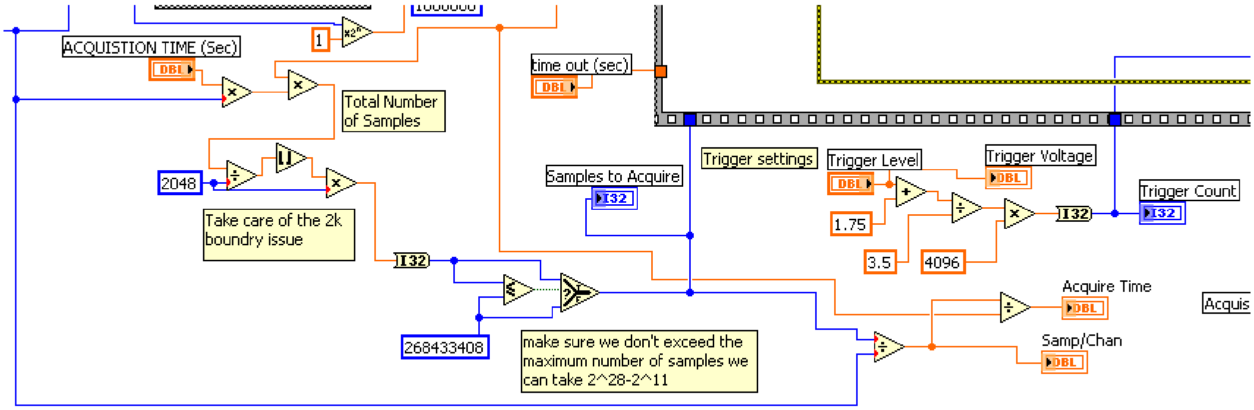


Figure 11. GUI initialization wiring diagram (part 2)

Figure 11 shows the remaining steps necessary to check that the total number of samples is not exceeded and to check the control that sets the trigger level for the Signatec boards.

LabVIEW structures, called frames, serve the same function as case statements in regular programming. This allows sections of the code to be executed based on the state of the software or in a specific order. Figure 12 shows frame 0, or the first frame that is executed. The mission is to configure all of the Signatec cards with the correct data and in the correct order.

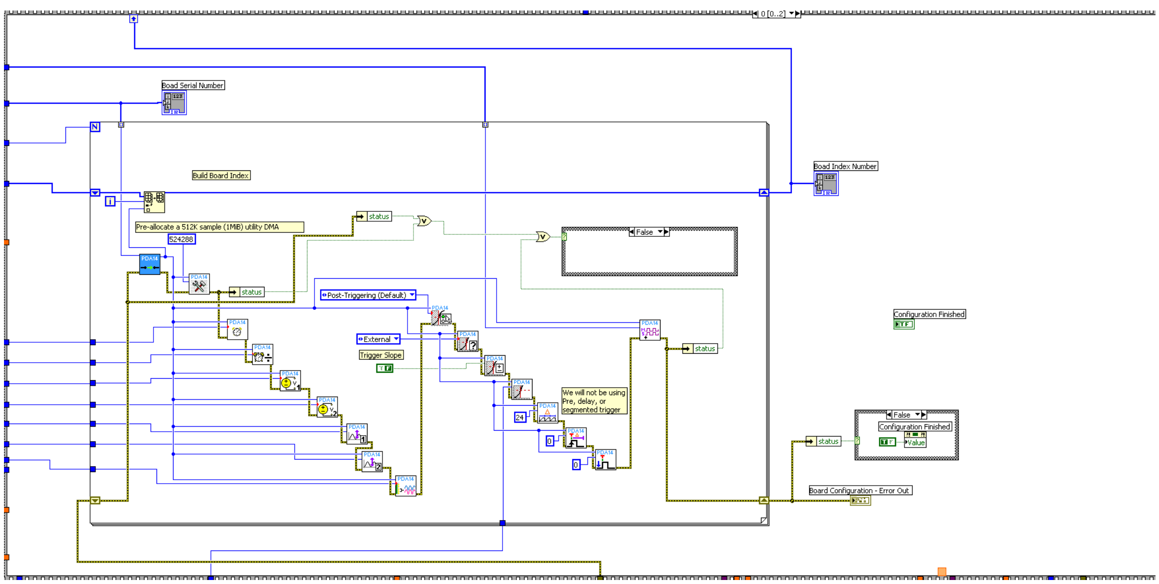


Figure 12. Wire diagram frame 0

The values flow from Figures 10 and 11 into the frame. The large internal block inside the frame represents a FOR LOOP to access each Signatec A/D card in turn. Each small block inside the FOR LOOP applies a specific configuration value to the Signatec card being addressed. This will include, but is not limited to the internal/external clock, clock rate, sample size, trigger level, master/slave configuration, and the input voltage range. Once all of the cards have been configured, the FOR LOOP is exited, and the processing continues to the next frame.

Figure 13 shows Frame 1, or the second frame. This frame contains a set of flat frames within it to make sure the program execution continues in a set order. From left to right, the first small frame sets a system flag that displays the "busy wait" mouse icon to show the user that the system is busy. The next frame is responsible for allocating the memory buffer on each card. The third frame sets the card to acquire to RAM or to buffer the acquired data until the end of acquisition. This frame also displays the current state of configuration registers of the Signatec cards in the "PDA 14 Before" area on the front panel.

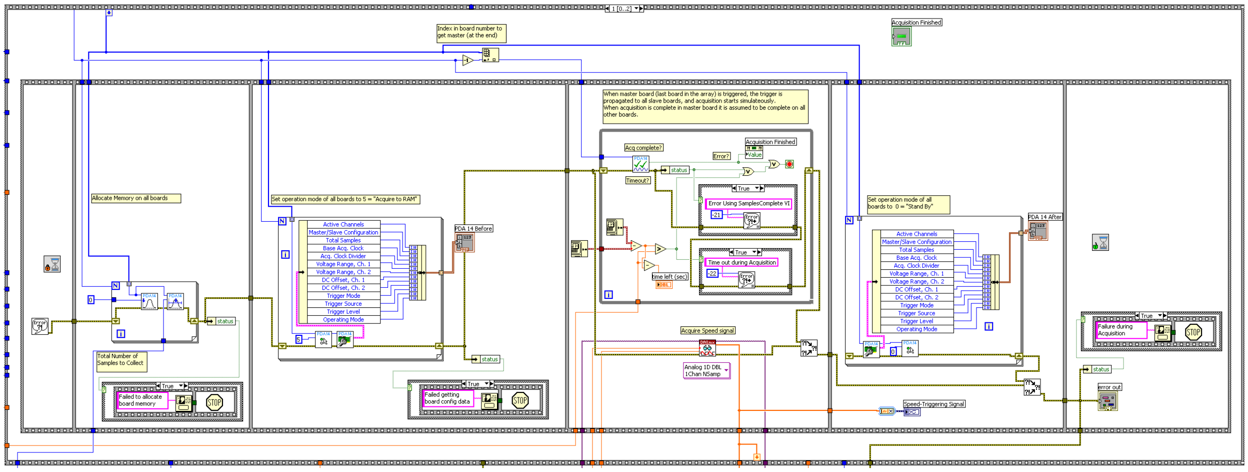


Figure 13. Wire diagram frame 1

The fourth frame shown in Figure 13 is responsible for waiting for the Signatec cards to be triggered and to finish acquisition. This frame has the start acquisition VI for the NI-6120 card. Also, a timer keeps track of the time remaining before the Signatec acquisition times out. This value is displayed on the front panel in the "Board Configuration" section in the "time left" indicator. The Signatec data acquisition has to be completed within this time window, or an error will be generated, and the program will stop. For this frame to finish, the Signatec cards must finish acquisition or time out, and the NI-6120 card must finish its acquisition cycle. When the Signatec and NI cards finish with their acquisition, the program continues to the next frame.

Once data acquisition is complete, the next frame in the series, Frame 5, reads the configuration registers of the PDA 15 and displays the data on the GUI front panel in the area labeled "PDA 14 After". The data displayed is mainly used for debugging purposes. Frame 6 resets the system flag to let the mouse cursor go back to normal, and the program flows to the next frame.

Figure 14 shows Frame 2, or the third frame. This frame is responsible for opening an output file and storing the data from the Signatec cards and the photocell data collected by the NI-6120. This frame checks the front panel control, "Ask to save data". If the "Ask to save data" control is set, then the software presents a dialog box and asks the user if he/she would like to save the data. If the answer is no, the program exits. If the answer is yes, a file is opened, the header section is written, the data is retrieved from each Signatec card, the photocell header is written, and the photocell data is written.

While the data is being retrieved from the cards and stored to disk, the program samples the data stream as specified by the "Start (CH sample)" and "Length (samples)" controls on the front

panel. When the system is finished saving the data, this data is presented to the user via the two large graphs on the lower right side of the GUI.

Also, while the data from each card is being downloaded, an indicator on the GUI is activated to show which card is currently being accessed to provide the user with some indication of progress, as this step takes several seconds to complete.

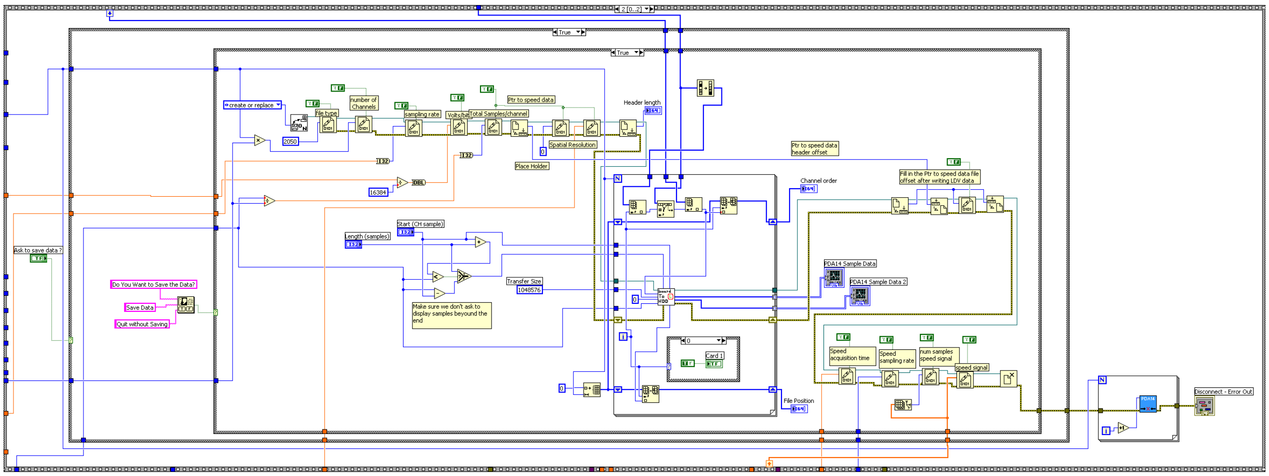


Figure 14. Wire diagram frame 2

Data Processing Discussion
User's Guide

All of the data processing uses Matlab after acquisition. The two main steps involve demodulation, which is done via the "TDMVelChan" function, and de-multiplexing, which is performed with the "TDMdmux" function.

To process a data file collected by the LabVIEW data acquisition software, one should call the "TDMVelChan" function with an argument of 0 (zero), which indicates that all channels will be processed, or the channel number of interest (i.e., 5 for channel five). A file selection GUI will be displayed, from which the input file is selected. Once the input file is chosen, another file selection GUI is opened so that the output file for the demodulated velocity data can be selected. The output file will have a default extension of vel. A progress bar is then displayed. After processing all of the data, the function returns the velocity data and an estimate of the intermediate frequency (IF) for each section of the data processed. This step can currently take several hours to complete depending upon the amount of data collected. It now takes about 20 minutes to process 0.06 seconds of data.

Once the velocity data file has been written, the velocity data can be de-multiplexed with the "TDMdmux" function to produce a velocity profile over a section of scanned ground. . The function can be called without any arguments or with the path and file name of the velocity file produced from the "TDMVelChan" function. If the function is called without arguments, a file selection GUI will be presented to select the appropriate velocity file. After the data is

processed, another file selection GUI opens to select the output file. The output file will have the default extension of vdm.

Program Flow – TDMVelChan

A simplified overview of the program flow of the function “TDMVelChan” is shown in Figure 15. The source code is provided in Appendix D. The main purpose of the “TDMVelChan” function is to demodulate the LDV data from the IF signal provided by the photodiodes and produce velocity data of the surface scanned by the lasers. The raw data is bandpass filtered, up sampled, mixed to produce I and Q, demodulated using the derivative of the arc tangent method to find the velocity, filtered, and down-sampled in three stages before being written to disk. This data processing step currently takes a long time, approximately 20 minutes for each 0.06 seconds of data. For a complete data set of 2.56 seconds, it takes in excess of 12 hours to process on a moderately fast computer.

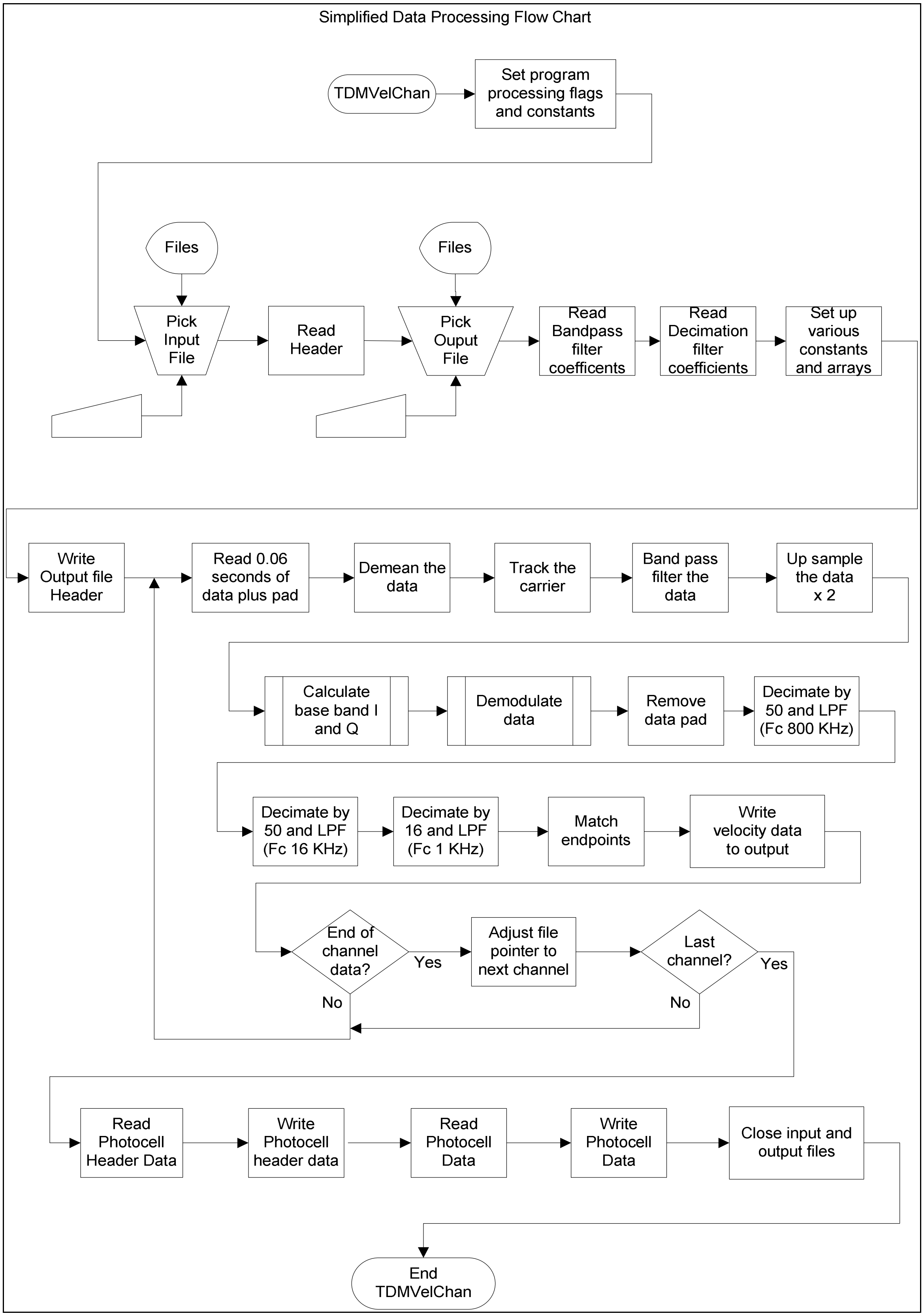


Figure 15. Simplified flow diagram for TDMVelChan

A major part of processing time is due to filtering and down-sampling the data. As shown in Figure 15, there are 4 filter blocks. Table 2 lists the characteristics of each of the filters. To minimize the end effects, the front and back of the data array is pre- and post-padded with data from the beginning and end before it is passed to the filter. The padding is removed after filtering. The input vector and filter coefficients are passed to the "filtfilt Matlab" function, which implements a zero-phase digital filter by processing the input in both the forward and reverse directions. This also has the effect of doubling the pass band ripple and the stop band attenuation. All of the filter coefficients are stored in files of the same name with a ".mat" extension. The coefficients for the filters are listed in Appendix E.

Filter Name	Type	Order	Fpass	Ripple	Fstop	Atten.
100Mfs_5_15Mbpf_fir45	Equiripple Bandpass	124	5 MHz 15 MHz	0.05 dB	3 MHz 17 MHz	-45 dB
100Mfs_800Klpf_fir45	Equiripple Lowpass	1233	800 kHz	0.05 dB	1 MHz	-45 dB
2Mfs_16klpf_fir45	Equiripple Lowpass	1233	16 kHz	0.05	20 kHz	-45 dB
40Kfs_1Klpf_fir45	Equiripple Lowpass	66	1kHz	0.05 dB	2.5 kHz	-45 dB

Table 2. Filter Characteristics

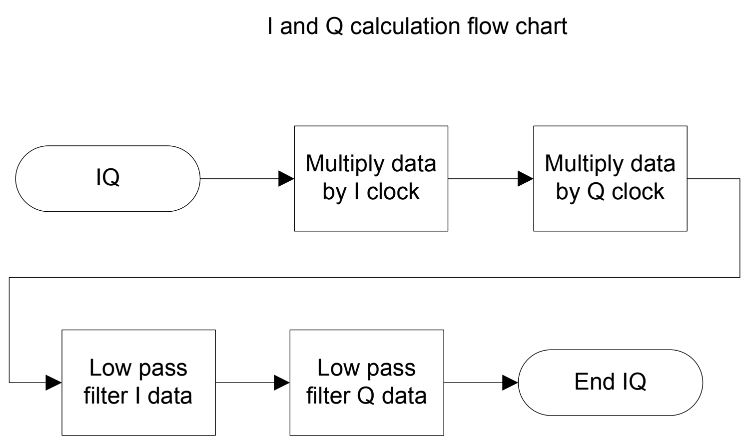


Figure 16. Simplified process flow for IQ function

Figure 16 shows the flow within the IQ function in which the base band I and Q components are calculated. This function is very straightforward as shown by Figure 16. The I and Q clock arrays can be either generated on a channel by channel basis or for each 0.06 second data segment. By setting the track flag (trackf) to the value of 2, the clock arrays are generated for each data segment. This allows for compensation of the variability of the Doppler offset due to vehicle speed variation. After the data and clock components are mixed, the I and Q data streams are passed through a low-pass filter to remove the non-base band components. The low-pass filter is implemented as an "Equiripple FIR filter" of order 226, with a pass frequency of 5 MHz and a stop frequency of 6 MHz. The pass-band ripple is 0.1 dB, and the stop band

attenuation is -45 dB. The Matlab function used to implement the filter performs a zero phase by processing the input in both the forward and reverse directions. This also has the effect of doubling the pass band ripple and the stop band attenuation. The coefficients for this filter are stored in the file “100Mfs_5Mlpf_fir45.mat” and are also listed as part of Appendix E.

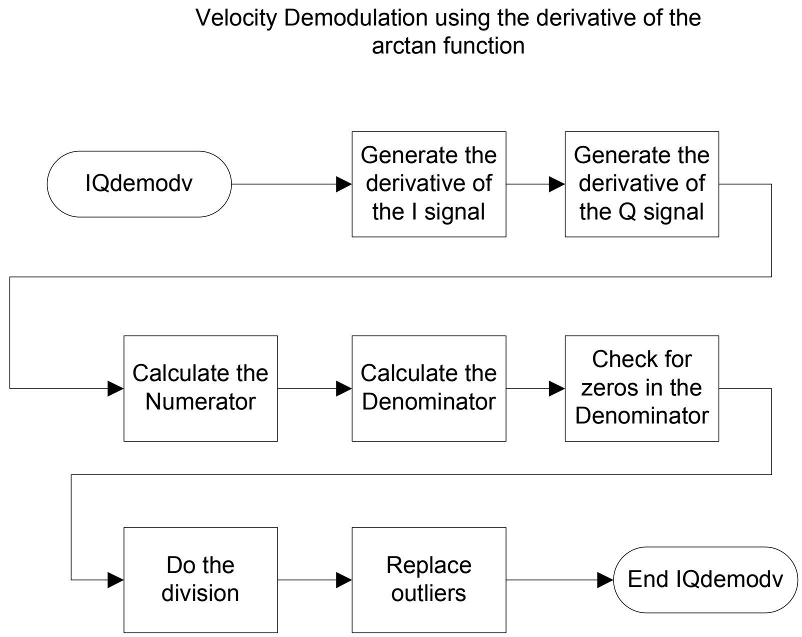


Figure 17. Simplified process flow for IQdemodv function

Figure 17 shows the flow with in the “IQdemodv” function which implements Equation 3, in which the velocity is calculated from the base band I and Q components. Lambda represents the wavelength of the laser, which is 532 nm for the current system.

$$v = \frac{\lambda}{4\pi}\left(\frac{IQ' - I'Q}{I^2 + Q^2}\right) \tag{3}$$

It is important to have the discreet derivative of the I and Q data point aligned with the original arrays. This means that, for a given I value, the Q derivative must correspond to the same sample point. For this reason, the mid-point technique, shown in Equation 4, is used to calculate the discrete derivative. This technique will ensure that the discrete derivative at point n will correspond to that point and not be a half sample off. Note that Δt represents the time between samples or the reciprocal of the sample rate.

$$x_n' = \frac{x_{n-1} + x_{n+1}}{2\Delta t} \tag{4}$$

The block “Replace outliers” is one of the non-linear filters that was implemented to deal with speckle noise. The noise spikes generally have a peak value that is much greater than the overall RMS value of the signal. The code generates a threshold value that is 10 times the RMS value of

the current data segment and then replaces any peak values greater than the threshold with the threshold value. This provides a minimal amount of signal-to-noise gain.

Table 3 provides a dependency list for the “TDMVelChan” function. The internal functions are contained within the “TDMVelChan” source file. The external functions are functions developed at NCPA. Documentation on all other functions is available from Matlab since they are native to Matlab.

Function Name	Description
IQ(iclk, qclk, sig, a, b)	Internal. Called from TDMVelChan This function returns the I and Q base band components of a signal. The arguments are the I and Q clock arrays (iclk and qclk), the signal vector (sig), the low pass filter coefficients (a, b).
IQdemodv(ib, qb, dt)	Internal. Called from TDMVelChan This function demodulates a frequency-modulated signal from its base band components. The arguments are the base band I component (ib), the base band Q component (qb), and the time between samples (dt). This function implements the derivative of the arc tangent demodulation function.
ZeroCk(d)	Internal. Called from TDMVelChan This function takes a data array and checks for values equal to zero. The zero values are replaced with a value of 1*10^-6 or the average of the value before and after.
findPeak(x, y)	External. Called from TDMVelChan This function will take the y vector and return the y values in descending order. If the x vector is included, the output can include an ordered array of x values that keep their correspondence to the original y vector.
fderiv(d)	External. Called from findPeak This function returns a discrete derivative using the mid-point technique from the values in d array.
zeroCross(d, type, flag)	External. Called from findPeak This function returns the locations of zero crossings in the data array. The arguments are the data array (d), the type of zero crossing – positive, negative, or both (type) – and a flag to select a filter type that reduces the extraneous number of zero crossings.

Table 3. TDMVelChan Function dependences

Program Flow – TDMdmux

A simplified overview of the program flow of the function “TDMdmux” is shown in Figure 18. The source code is provided in the Source Code Appendix. The main purpose of the “TDMdmux” function is to extract the pieces of velocity time series from each LDV channel that corresponds to a common ground segment, thereby de-multiplexing the surface velocity information from LDV signals.

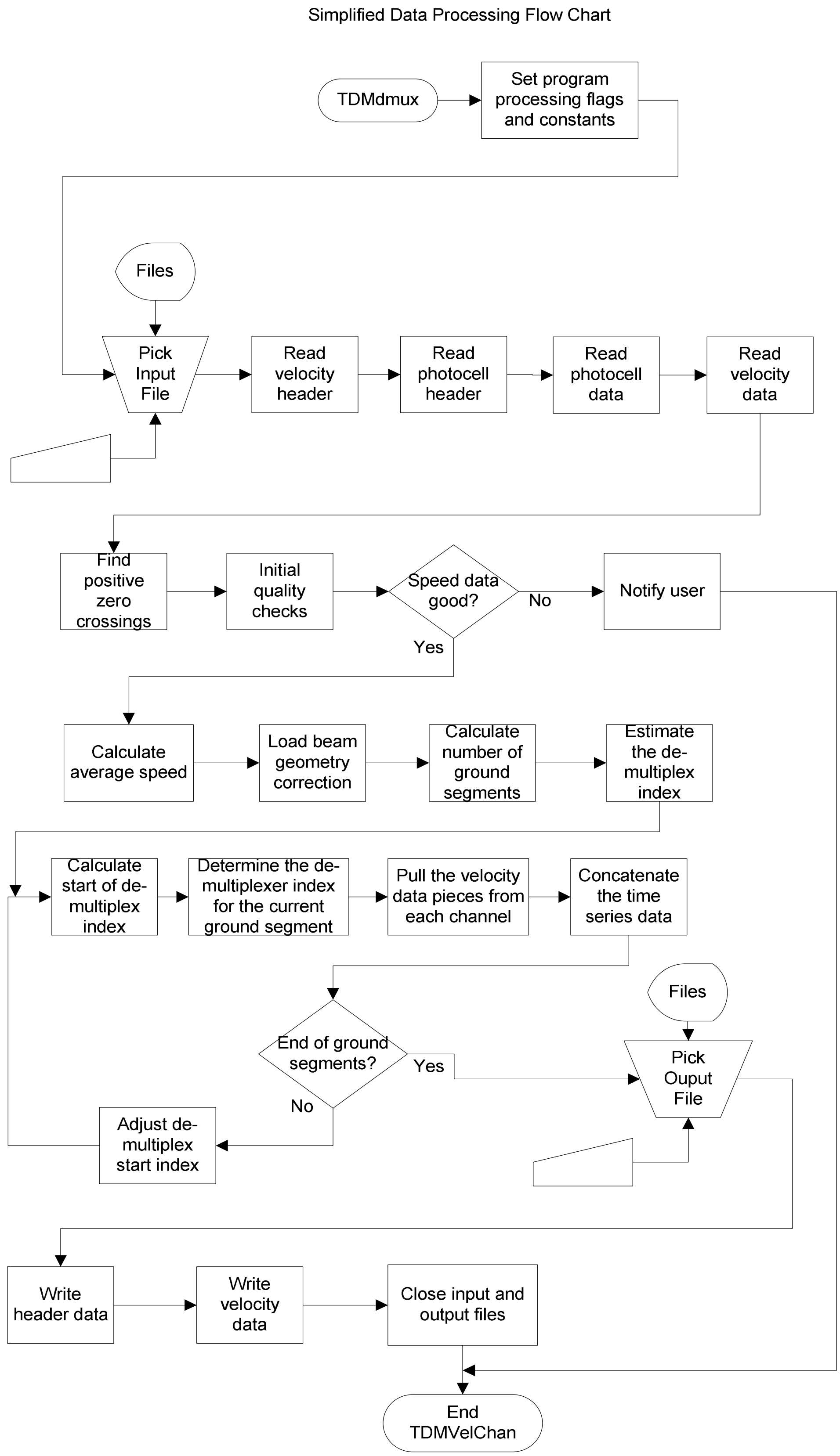


Figure 18. Simplified process flow for TDMdmux

Table 4 provides a dependency list for the "TDMdmux" function. The external functions are Matlab functions developed within NCPA. Documentation on all other functions is available from Matlab.

Function Name	Description
zeroCross(d, type, flag)	External. Called from TDMdmux This function returns the locations of zero crossings in the data array. The arguments are the data array (d), the type of zero crossing – positive, negative, or both (type) – and a flag to select a filter type that reduces the extraneous number of zero crossings.

Table 4. TDMdmux Function Dependencies

Experimental Results

UM was able to take the electric vehicle, LDV, and data acquisition system and collect data in the laboratory. However, due to the lack of sensitivity of the photodiode electronics, the test track surfaces had to be prepared by either having glass beads on a sand surface or by embedding the glass beads in clear latex adhesive on wooden planks. The test track used for the data shown in this report is shown schematically in Figure 19. The 2x4 planks were placed on lead bricks to isolate them from any vibration from the lab floor. The lead bricks also provided the support necessary to make the off target surface close to the same height as the vibrating plate.

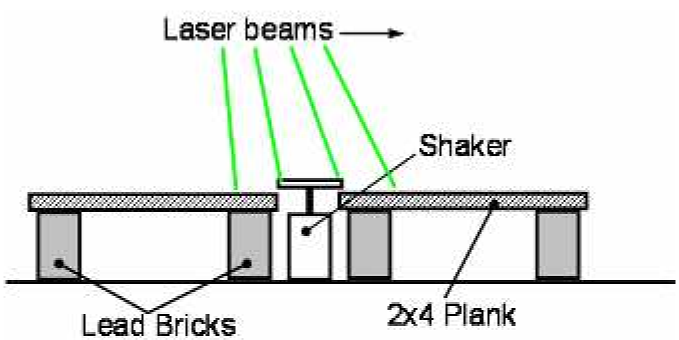


Figure 19. Test track using 2x4 planks and shaker

The plate was made larger than the beam spacing to allow two sets of beams to see the plate at the same time. The LDV was moved over the test track, and the data was processed. Figures 20 and 21 show the demodulated velocity data from each beam. Beam 1 had a bad photo detector, so it is not shown in the plots. Referring back to Figure 2, Beam 2 was the beam furthest away and has the largest angular displacement from vertical, while Beam 16 was the beam closest to the vehicle and was perpendicular to the ground. The time series shows that a slight overlap existed between one beam going off target and the next beam just starting to be on target. This can be seen in the demodulated time series for each beam.

Figures 20 and 21 clearly show the progression of each beam over the target. The shaker signal progressed from Beam 2 to Beam 3 and so forth. This shows the target signal being time-

multiplexed onto each beam as the LDV moved down the track. By being able to combine each section of each beam as it passed over the target, processing gains were made by analyzing that composite signal.

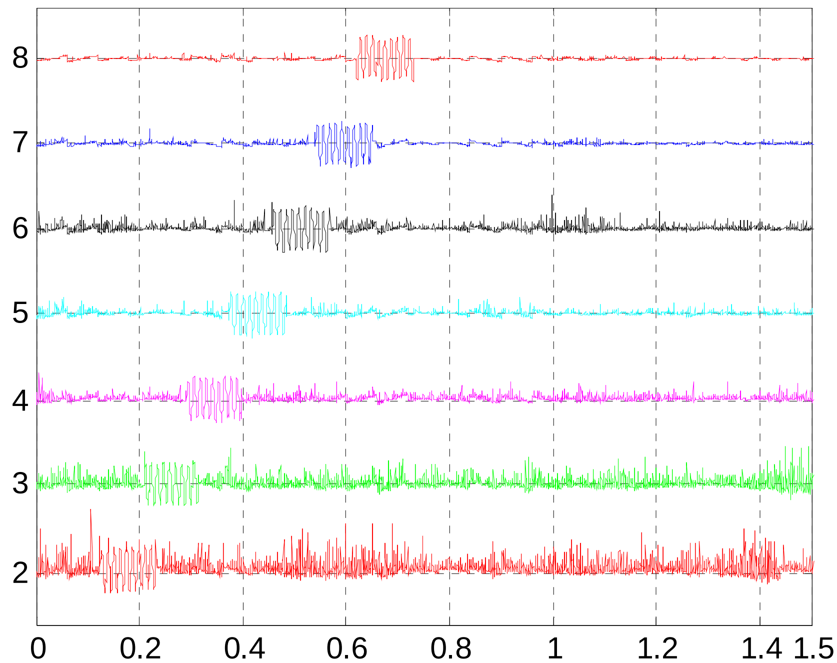


Figure20. Time Series for channels 2 through 8

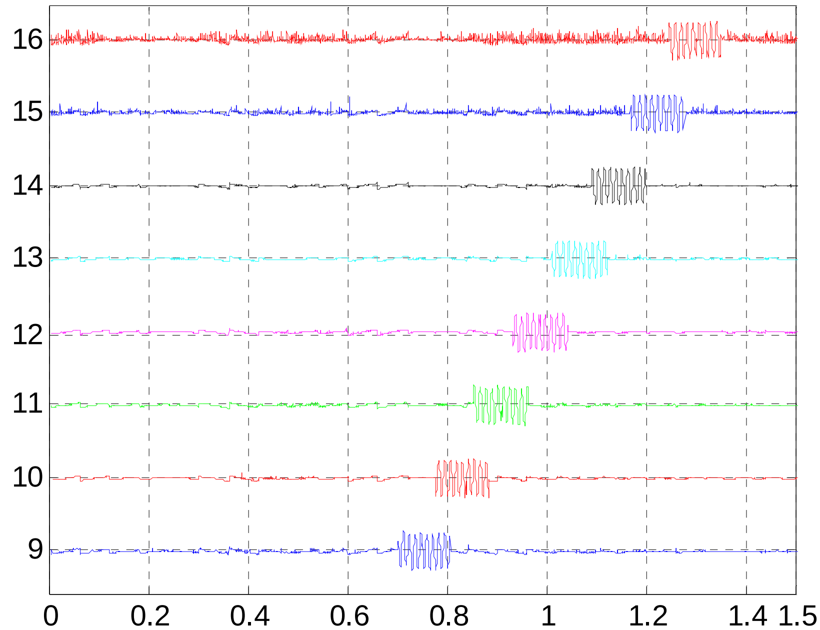


Figure 21. Time series for channels 9 through 16

The limitation of the current photodetectors and the need to improve the surface reflectivity of test tracks with glass beads can be seen in Figure 20. Note that Beam 2, which has the largest angle from vertical, has the most noise. However, inspection of the part of the time series where the beam passed over the vibrating plate shows that the noise was much less than the signal, as shown in Figure 22. This discrepancy is due to the reflectivity difference between the two surfaces. Although the surfaces were prepared similarly, the vibrating plate had a higher intrinsic reflectivity. Therefore, more laser light was reflected back to the sensor, as seen in the raw data from the photo diode, in which the carrier signal was much less prominent over the 2x4 lumber but distinctly visible over the plate. Signals without the carrier indicate that the photodiode was detecting mostly noise, not the laser light from the surface, which then demodulated to a signal that has much higher noise statistics.

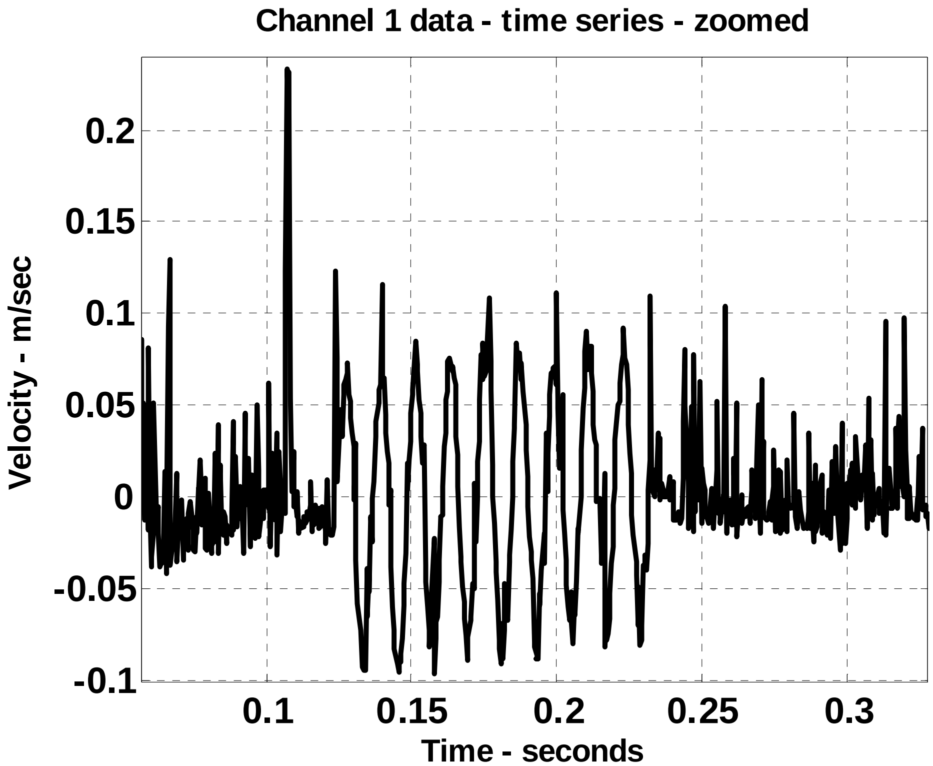


Figure 22. Beam 2 zoomed to passage over shaker

As seen in Figure 3, UM was able to take data, demodulate the data to get the surface velocity signal, and then demultiplex the data to form a time series that was 16 times longer than the dwell times associated with individual LDV beams. The shaker was set for a vibrational velocity of 35 mm/sec. Also seen in Figure 3, the recovered velocity only measured about 16.5 mm/sec. This difference is due to limitations in the demultiplexing algorithm and methods used to derive vehicle position and speed. The vehicle position and speed measurements are fairly crude, with the relative position being based on the spacing of the retro reflective tape at 10 cm increments. Likewise, the average velocity estimate is only known based on the same 10 cm spacing and the accuracy of the A/D timing. Figure 4 shows a much better estimate of the target velocity, using a different demultiplexing technique based on the correlation of adjacent beams to derive the offset index. This technique shows a velocity measurement of 34 cm/sec. The time series from this method was much better, with fewer gaps and discontinuities. As seen in Figure 23, the gaps in the demultiplexed time series result in less energy being attributed to the shaker. These gaps also produced the side lobes that appear in the demultiplexed signal shown in Figure 3. Figure 24 shows the time series that resulted from using the cross correlation method.

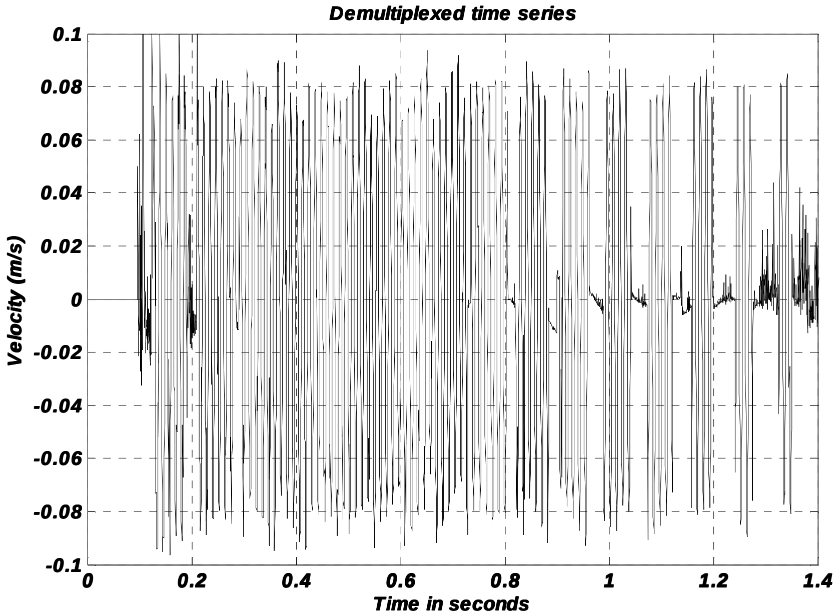


Figure 23. Auto demultiplexed time

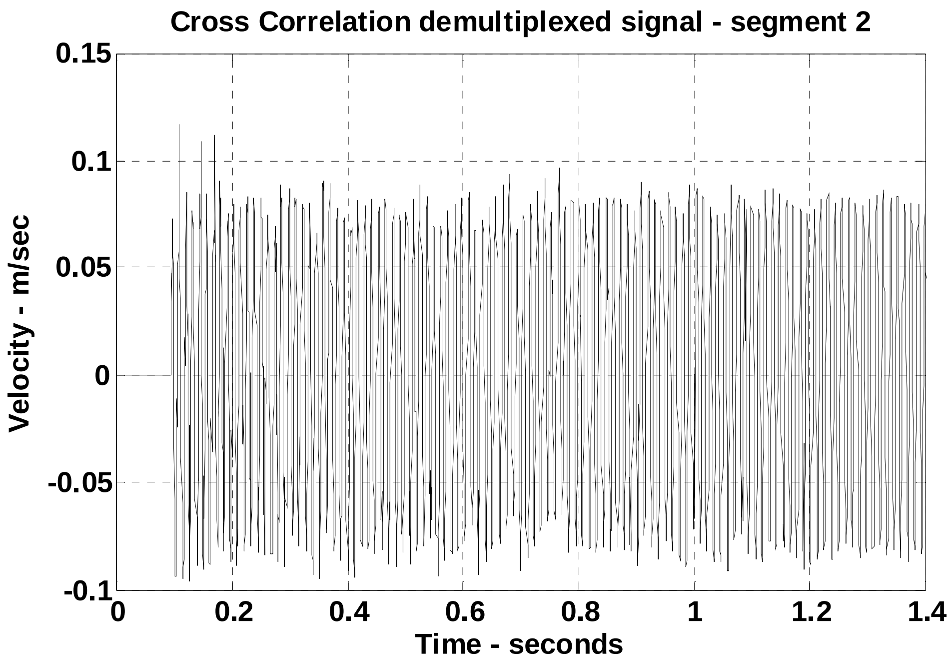


Figure 24. Manual demultiplexing using cross correlation

Based on Figures 3 and 4, along with Figures 22 and 23, UM has met the basic goal of being able to demonstrate the time division multiplexing concept using a multibeam LDV on a moving platform. However, during this development and implementation phase, several system limitations became apparent that require additional work.

Future work

Items to be addressed in future research include:

- Implementing the demultiplexing algorithms in real time using dedicated hardware. This will allow the system to perform continuous scanning.
- Optimizing the photodetector electronics for better sensitivity and improving the SNR to work on realistic ground surfaces.
- Improving the demultiplexing by using measured position and speed instead of estimates of position and speed of the vehicle. The expected signal gain is shown by comparing Figure 4 with Figure 3.
- Developing and implementing the coupled motion algorithms using differential vibrometry.
- Improving the SNR by determining the appropriate mixture of speckle mitigation algorithms
- Obtaining a velocity profile of a buried vibrating target at a field site with prepared surface
- Obtaining a velocity profile of a buried vibrating target at a field site with a normal surface

Appendix A

Program Statistics

(1) Submissions or publications under ARO sponsorship. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals – 0

(b) Papers published in non-peer-reviewed journals – 0

(c) Presentations

(i) Presentations at meetings, but not published in Conference Proceedings – 5

Richard B. Burgett, Vyacheslav Aranchuk, James M. Sabatier, "Mitigation of speckle noise due to laser Doppler vibrometer motion across a vibrating target," J. Acoust. Soc. Am., 124, 2500 (2008).

Vyacheslav Aranchuk, James M. Sabatier, and Richard D. Burgett, "Fast acoustic landmine detection using multiple beam laser Doppler vibrometry," J. Acoust. Soc. Am. 123, 3044 (2008).

James M. Sabatier, Vyacheslav Aranchuk, Richard D. Burgett, "Increased Ground Vibration Measurement Speed for Landmine Detection," 12th Landmine and Explosive Object Detection Research Review Meeting, January 2009.

James M. Sabatier, Vyacheslav Aranchuk, Richard D. Burgett, "Buried Landmine Velocity Profile with a Time-Multiplexed LDV," 11th Annual Landmine Detection Research Review Meeting, January 2008.

James M. Sabatier, "High-Speed Vibrometry for Acoustic Landmine Detection," Review of the ARO Terrestrial Sciences Program October 2007.

(ii) Non-peer-reviewed Conference Proceeding publications – 1

Richard Burgett, Vyacheslav Aranchuk, James M. Sabatier, Stephen S. Bishop, "Demultiplexing multiple-beam laser Doppler vibrometry for continuous scanning," Proc. SPIE Vol 7303, DOI 10.1117/12.818219 (2009).

(iii) Peer-reviewed Conference Proceeding publications – 0

(d) Manuscripts – 0

(e) Books – 0

(f) Honors and awards – 0

(g) Titles of Patents Disclosed during the reporting period – 1

James M. Sabatier, Vyacheslav Aranchuk, and Richard D. Burgett, “High-speed vibration measurement by multiplexing channels of a continuously moving multiple beam laser Doppler vibrometer,” Research Disclosure to the University of Mississippi.

(h) Patents awarded during the reporting period – 0

(2) Student/Supported Personnel Metrics**.**

(a) Graduate Students -2

(b) Post Doctorates - 0.

(c) Faculty -0.

(d) Undergraduate Students -0.

(e) Graduating Undergraduate Metrics – Not applicable

(f) Master’s degrees awarded – 1

(g) Ph.D.s awarded – 0

(h) Other research staff –
(i) James M. Sabatier, Principal Scientist, 5% FTE
(ii) Vyacheslav Aranchuk, Senior Research Scientist, 5% FTE
(iii) Richard Mack, Associate Research & Development Engineer, 8% FTE
(iv) Ina Aranchuk, Associate Research & Development Engineer, 35% FTE

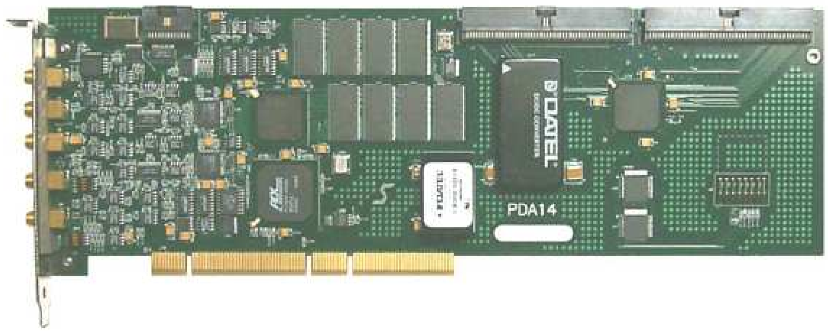
(3) Technology Transfer

Research results have been presented at the US Army Night Vision and Electronic Sensors Directorate to Mr. Steve Bishop. Results have also been presented at ARO program reviews.

Product Information Sheet

PDA14

2 Channel, 14-Bit Waveform Digitizer



FEATURES

- 2 Channels at up to 100 MHz Sample Rate
- 14 Bits of Resolution
- Bandwidth from DC-50 MHz
- 512 Megabytes of On-Board Memory
- 500 MB/s Transfer via Signatec Auxiliary Bus (SAB)
- 266 MB/s Transfer (Peak) Over PCI Bus
- 64/32 Bit PCI Plug and Play Compatible Board

APPLICATIONS

- Radar
- Mass Spectroscopy
- Mass Spectrometry – Time of Flight
- Communications
- Ultrasound
  Medical Diagnostics / Non Destructive Testing
- Laser Doppler Velocimetry
- High Speed Waveform Capture

OVERVIEW

The PDA14 is a dual channel waveform capture board which provides a tremendous combination of high speed and high resolution along with an extremely large memory capacity. The entire 512 MB memory may be used as a giant FIFO for acquiring data directly to either the SAB or PCI bus. Tests have shown that a very large FIFO memory is required to prevent data loss when performing continuous data streaming over the PCI bus.

The PDA14 is a 64-bit PCI compatible board equipped with standard 'Plug and Play' features common in PCI systems. It is capable of Bus Master DMA data transfers at up to a peak of 266 megabytes/second and a sustained rate of 250 megabytes/second. It can also operate in 32-bit PCI slots in which case the transfer rate will be limited to about 125 megabytes per second.

The PDA14 incorporates the advanced Signatec Auxiliary Bus (SAB) that allows for data transfers of up to 500 megabytes/second. This allows for the high-speed transfer of data to fast processor boards, such as Signatec's PMP8A, or other peripherals. The SAB also incorporates device control features for operating the PDA14 independent of the host bus.

The PDA14 is equipped with an interconnect port to allow multiple boards to be interconnected in a Master/Slave configuration. Up to three Slave boards may be operated with one Master. Master/Slave connections are via a ribbon cable that connects at the top of the board. In this configuration the clock and trigger signals from the Master drive the Slave boards so that data sampling on all boards occurs simultaneously.

The PDA14 has six software selectable signal amplitude ranges from a maximum of 3.0 volts down to 200 millivolts full scale.

External clock and trigger signals are provided via SMA connectors on the back bracket. Also provided is a user selectable digital output signal for synchronization purposes. Effectively, twenty-two internal clock frequencies may be selected, from 100 MHz down to 97.7 kHz in factors of 2 or from 62.5 MHz down 61.0 kHz in factors of 2. The PDA14 supports single shot, segmented, and pretrigger triggering modes.

Signatec, Inc. • 359 San Miguel Drive, Suite 300 • Newport Beach, CA 92660 USA • Phone (949) 729-1084 • Fax (949) 729-9114 • http://www.signatec.com

HARDWARE DESCRIPTION

The figure on the next page shows a simplified mechanization for the PDA14. The input signals may be set for either DC or AC coupling. The signal conditioning provides six voltage ranges from 200 millivolts to 3.0 volts peak-to-peak full scale. A low pass filter sets the channel bandwidth to 50 MHz. For test purposes, the ADC input may be fed a sine wave test signal (not shown) for measuring the AC performance at 25 MHz.

The data input to RAM FIFO 1 can be considered to be two separate 16-bit data streams. The latches and multiplexers at the output of the ADCs allows for capturing data from both channels or from channel 1 only or channel 2 only.

The Pretrigger Samples Shift Registers are programmable in length up to 4k samples. They can be used to capture pretrigger samples in either the Single Shot or Segmented trigger modes. Before a trigger is received, data is written into the shift registers but goes no further. After receiving a trigger, data samples start to be written into RAM FIFO 1. See the section "Trigger Modes and Options" for trigger mode details.

The ADCs always operate at either 100 MHz or 62.5 MHz when the internal clock is used or else at the external clock frequency. Operation at reduced rates is accomplished by dropping out the appropriate data bytes from the data stream. Thus the effective sample rate is divided from 2 to 1024 in factors of 2.

Data is written into the SDRAM via FIFO1 and read from RAM via FIFO2. The RAM operates at a clock rate of 133 MHz so it has a bandwidth capability of slightly greater than 500 MB/s. When data is being acquired at a rate of 250 MB/s or less (62.5 MHz on 2 channels or 100 MHz on single channel), it is possible to operate the board in Buffered Acquisition mode. In this mode the RAM is operated as a very large FIFO for acquiring data directly to the PCI bus or SAB.

The PDA14 has 12 operating modes as follows:

- Standby
- Acquisition to RAM
- Acquisition to PCI Bus
- Buffered Acquisition to PCI bus (RAM as FIFO)
- Acquisition to SAB
- Buffered Acquisition to SAB (RAM as FIFO)
- Data Transfer, RAM to PCI bus
- Data Transfer, RAM to SAB
- Write RAM (from PCI bus)
- Write RAM (from SAB)
- Read Time Stamps (from PCI bus)
- Read Time Stamps (from SAB)

The RAM write mode is typically used to test the on-board memory by writing data via the PCI bus and reading back via a RAM-to-PCI transfer.

External Inputs/Outputs

Besides the signal input, the PDA14 also provides SMA connections for a clock input, a trigger input, and a digital output signal. The clock input can be selected as the ADC sample clock. The external trigger can be used to synchronize the start of data acquisition with an external event. Trigger parameters such as trigger level, slope, etc. are user programmable. The digital output is a user selectable signal. The list of selections is TBD.

Trigger Modes and Options

In data acquisition mode three triggering modes are available: single shot, segmented, or pretrigger. In the single shot mode, following the detection of a trigger signal, all of the active memory is filled. In the segmented mode a separate trigger signal is required to successively fill each memory segment until all of the active memory is filled. In the pretrigger mode the board is armed and continuously fills the entire active memory until a stop trigger is detected, after which a programmed number of post trigger samples are taken before acquisition is terminated. The pretrigger mode may be used to see signal information both before and after the trigger signal.

Samples Settings

There are several board settings that affect the quantity and method of acquiring samples.

Active Memory Size – In the "post-trigger modes" this is the number of samples that will be taken after which the memory will be considered "full" and the acquisition is terminated. When a full condition is detected, a flag is set which may be read by the PC or software selected to cause a PC interrupt or send an interrupt over the SAB. The amount of memory that is activated for data acquisition may be set from 8 bytes to the full 512 megabytes in steps of 8 bytes.

Segment Size – In Segmented Mode this is the number of samples that will be taken each time a valid trigger signal is detected.

Pretrigger Samples – In Single Shot or Segmented Modes, this is the number of samples that will be recorded into RAM that occurred before the trigger.

Delayed Trigger – This sets a delay between the actual applied trigger and the effective trigger for the board. The delay range is from 0 to 64k digitizer clock cycles. In Pretrigger Samples mode the delayed trigger setting establishes the number of post-trigger samples that will be recorded.

Time Stamps

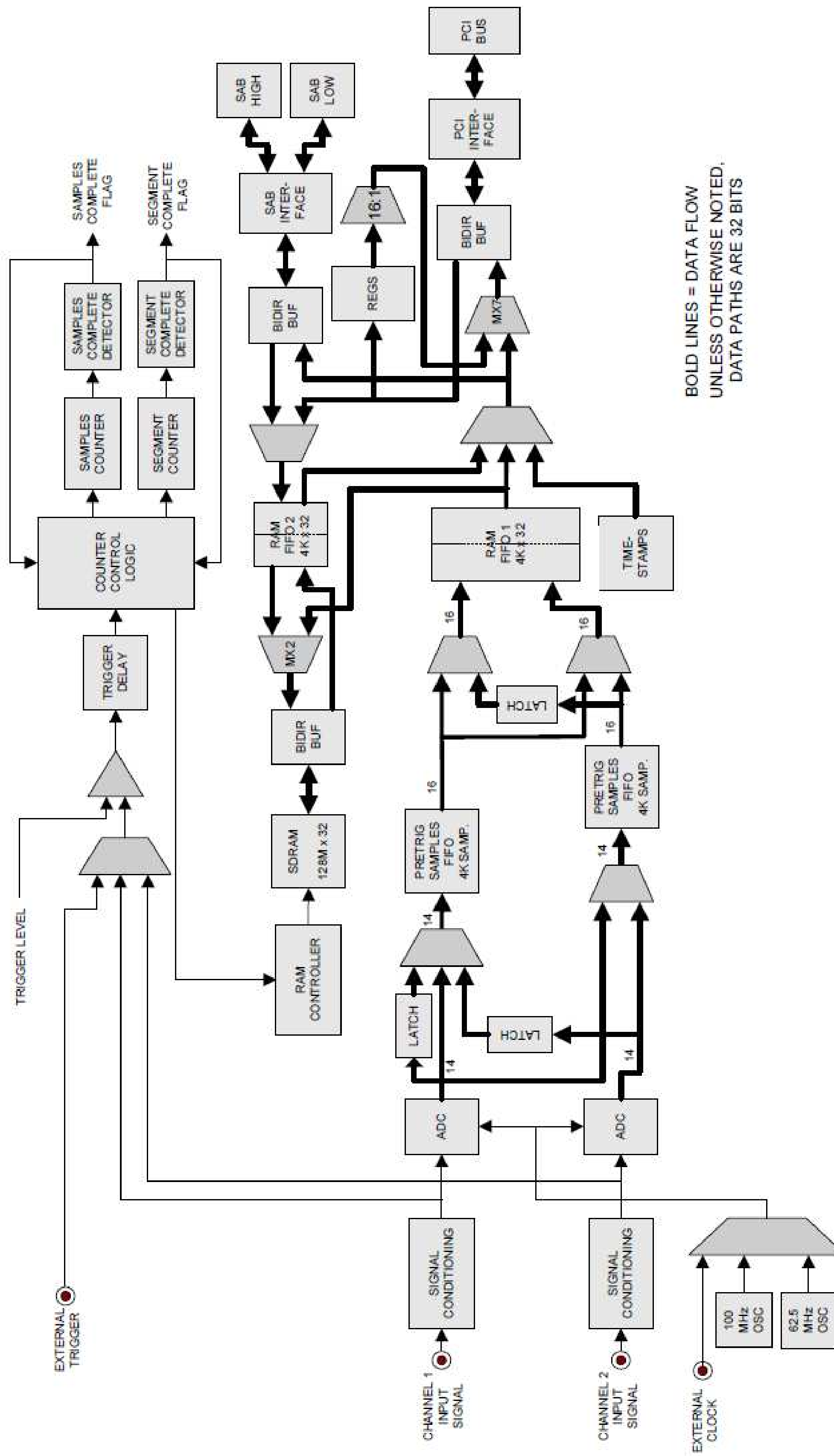
In Segmented Mode "time stamps" allow for storing the time relationship between the memory segments. Time Stamps are 32 bit timer values with a clock resolution of 7.5 nanoseconds. Up to 2048 time stamps are accumulated in memory separate from the data. Time stamps are read after the acquisition is completed.

SAB Operation

The PDA14 can perform SAB data transfers at 64 bits, or at 32 bits over either the high (SABH) or low (SABL) bus ports. This provides flexibility when multiple boards are incorporated into a system. At 64 bits the maximum transfer rate is 500 MB/s.

PCI Operation

The PDA14 is capable of sustaining a long-term data-transfer rate, over the PCI bus, of 250 megabytes per second when installed in a 64 bit PCI slot. It can also be installed in traditional 32-bit slots in which case the maximum rate is about 125 megabytes per second.



PDA14 FUNCTIONALALITY

SOFTWARE, SYSTEM, AND PERFORMANCE DETAILS

Software

The PDA14 is supplied with the following software:

- Windows NT/2000/XP and Linux Drivers
- C Function Library with source code
- PDA14 Class (C++) that wraps the Library functions and handles the tedious initializations and basic operations.
- Software manual that describes how to use the available library of functions to create larger applications or systems.
- A board diagnostics self test program.
- Multiple Coding examples
  - ✓ Acquire with DMA transfer to PC
  - ✓ Digital oscilloscope interface software
  - ✓ Using multiple PDA14 boards (with Master/Slave support)
  - ✓ SAB data transfer examples

Maestro support for the PDA14

- Menu driven board settings for multiple PDA14 boards
- Using the PDA14 as a high-speed recording system
- Using the PDA14 as a high-speed recording system with the PMP8A (or any other SAB Active Slave capable device)
- Data Analysis tools for poking/peeking onboard memory

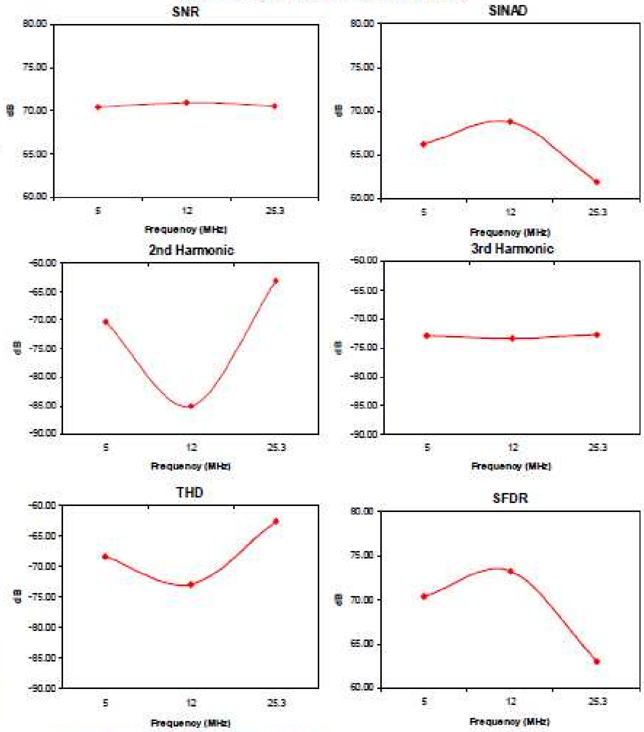
System Capabilities

The system solution offered by Signatec is based on supplying a comprehensive range of products incorporating the Signatec Auxiliary Bus. This 64-bit bus provides transfer rates up to 500 MB/s. SAB boards act as modular building blocks for constructing high performance systems that mechanize a wide variety of applications. Shown in the figure below are elements of such a system. Many systems can be constructed using standard desktop PC's. For demanding applications Signatec can supply a total turnkey system utilizing one of our industrial computer systems.

Devices connected to the SAB may communicate via SAB interrupt and control lines. This allows the boards to accomplish multiple acquisition, transfer, and processing cycles under control of the signal processor device, without PC intervention. Bypassing the host bus and operating system can significantly improve system performance.

Typical Performance

(dB versus Frequency in MHz)



DEFINITION OF TERMS

SINAD: Signal to Noise and Distortion: The ratio of the fundamental sinusoidal signal power to the total noise and distortion component power. In other words this is the ratio of the fundamental signal power to the measured power from the remainder of the detectable spectrum from dc to 50 MHz.

SNR: Signal to Noise Ratio: The ratio of the fundamental sinusoidal signal power to the noise power. For this data sheet noise is considered to be the power from all spectral components except for the fundamental signal, the first harmonic, and the second harmonic.

THD: Total Harmonic Distortion: The ratio of the total power of the second and third harmonics to the fundamental sinusoidal power.

Second Harmonic Distortion: The ratio of the power at twice the fundamental frequency to the power of the fundamental sinusoid.

Third Harmonic Distortion: The ratio of the power at three times the fundamental frequency to the power of the fundamental sinusoid.

SFDR: Spurious Free Dynamic Range: The ratio of the fundamental sinusoidal power to the power of the next highest spurious signal. Normally the highest spurious signal is the second or third harmonic.

TEST METHOD

A filtered sine wave signal is applied to the channel 1 and channel 2 inputs. Test frequencies used are 5.0, 12.0, and 25.3 MHz. The digitizer clock setting is 100 MHz. The voltage range is 600mV. Signal amplitude is set for 95% of full scale. Performance measurements are made using a 4096 point FFT with a Blackman-Harris window. Signatec uses the first 10 bins to represent the DC term, 64 bins centered around the peak for the fundamental signal power, 9 bins centered at twice the fundamental for the second harmonic and 9 bins centered at three times the fundamental for the third harmonic. All other bins are considered to be noise. (NOTE: large number of bins for the fundamental is necessary so that energy in the side lobes of the window function is not misinterpreted as noise or spurs.)

PDA14 SPECIFICATIONS AND ORDERING INFORMATION

External Signal Connections (SMA)
Analog Input, Channel 1
Analog Input, Channel 2
Clock Input
Trigger Input
Digital Output

Analog Inputs
Full Scale Volt. Ranges: 200mV, 333mV, 600mV, 1.00V, 1.66V, 3.00V
Impedance : 50 ohms
Bandwidth : 50 MHz
Equivalent Noise : 0.5 lsb RMS (typical)
Coupling : AC or DC[1]

External Trigger
Impedance : 1k ohms
Trigger Level : ±1.75 Volts
Adjustment Method : via 12 bit DAC
Bandwidth : 50 MHz
Coupling : DC

External Clock
Signal Type : sine wave or square wave
Coupling : AC
Impedance : 50 ohms
Frequency : 30 MHz to 100 MHz
Amplitude : 100 mV p-p to 2.0 V p-p

Digital Output
Type : TTL Logic Level
Max. Frequency : 100 MHz
Suggested Load : 1k ohms
Amplitude : TTL

DC Offset Voltage
Resolution: : 12 bit DAC
Range : ± 1.1 x FS ADC input

Digitizer
Voltage Range : 2.0V p-p full scale
Resolution : 14 bits
Linearity, Integral : ±0.5 lsb max.
Linearity, Differential : ±0.75 lsb max.
Aperture Jitter : < 0.5 pS typical

Internal Clock
Available Rates : 100 MHz down to 97.6 kHz in factors of 2
62.5 MHz down to 61.0 kHz in factors of 2
Accuracy : ± 0.01%

Trigger Modes
Post Trigger : single start trigger fills active memory
Pre-trigger : single trigger stops acquisition
Segmented : start trigger for each memory segment

Trigger Options
Pre-trigger Samples : samples prior to trigger are stored; Single Channel: 8k max.; Dual Channel: 4k max per channel
Delayed Trigger : delay from trigger to data storage; Up to 64k digitizer clock cycles

Memory
Active Size : Up to 256 MegaSamples
Segment Size : Up to 128 Megasamples
Start Address Setting : Anywhere in memory
Segment re-arm time[2] : 150 nanoseconds
Addressing : DMA transfer from starting address
Memory Address (PC) : Plug and Play selected

I/O Addressing
PCI Controller Address : 64 bytes, Plug and Play selected
Control/Status Registers: 32 bytes, Plug and Play selected

Signatec Auxiliary Bus
Data Transfer Modes : Block or Packet
Data Transfer Rates : 500 MB/s max @ 64 bits
Data Direction : output only

Power Requirements
+12V : 400mV Amps max.
+5V : 1.5 Amps max.
+3.3V : 2.3 Amps max.

Absolute Maximum Ratings
Analog Inputs : ±5 volts
Trigger Input : ±5 volts
Clock Input : 5 volts peak to peak
Ambient Temperature : 0 to 50°C

PDA14 Board
Part Number: PDA14

SAB Cables
Refer to the "SAB Cable Assembly Ordering Guide" to select and order the appropriate cable assemblies.

Master-Slave Cables
The PDA14 may be software configured to operate as a Master or a Slave in a multiple board system. In order to operate in a Master/Slave configuration a 20-pin ribbon cable is required to connect the boards. This cable is ordered using the basic part number PDA14MS-X where X is the total number of boards connected together. Master/Slave boards must occupy adjacent slots. The maximum number of boards to be connected is one master and three slaves.

Documentation & Accessories
The PDA14 is supplied with a comprehensive operator's manual, which thoroughly describes the operation of both the hardware and the software. Also supplied are two four-foot coaxial cables with BNC to SMA connectors. Extra cables may be purchased from Signatec. Supplied software disks contain a function library for Microsoft Visual C/C++, example programs, and all source code to libraries and examples.

Customer Support
The Signatec Web Site: www.signatec.com is the primary access point for software updates, documentation updates, or technical support. For the best technical support it is very important to follow the instructions on the technical support page.

Product Warranty
All Signatec products carry a full 3-year warranty. During the warranty period, Signatec will repair or replace any defective product at no cost to the customer. This warranty does not cover customer misuse or abuse of the products or physical damage not reported within 15 days of the time of shipment by Signatec.

Notes:
1. Selected via dip switch.
2. In segmented mode, time from the end of a segment until a trigger will be accepted to begin another segment acquisition.

Signatec reserves the right to make changes in this specification at any time without notice. The information furnished herein is believed to be accurate, however no responsibility is assumed for its use.

Data Sheet Revision 1.00 - Date 05-16-2005

NI 6115/6120 Specifications

This document lists the I/O terminal summary and specifications for the NI 6115/6120.

For the most current edition of this document, refer to `ni.com/manuals`. Refer to the *DAQ Quick Start Guide* for more information about accessing documents on the NI-DAQ CD.

Note With NI-DAQmx, National Instruments has revised its terminal names so they are easier to understand and more consistent among NI hardware and software products. The revised terminal names used in this document are usually similar to the names they replace. For a complete list of Traditional NI-DAQ terminal names and their NI-DAQmx equivalents, refer to the *Terminal Name Equivalents* table in the *S Series Help*.

Table 1. I/O Terminal Summary

Terminal Name	Terminal Type and Direction	Impedance Input/ Output	Protection (Volts) On/Off	Source (mA at V)	Sink (mA at V)	Rise Time (ns)	Bias
AI <0..3> + (NI 6115 only)	AI	100 MΩ in parallel with 10 pF[1] or 10 kΩ in parallel with 40 pF[2]	42 V	—	—	—	±300 pA
AI <0..3> – (NI 6115 only)	AI	10 nF to AI <0..3> GND	42 V	—	—	—	±300 pA
AI <0..3> + (NI 6120 only)	AI	100 GΩ to GND	±42 V to GND	—	—	—	±300 pA
AI <0..3> – (NI 6120 only)	AI	10 nF to AI <0..3> GND	±42 V to GND	—	—	—	±200 pA
Differential pair AI <0..3> + to AI <0..3> – (NI 6120 only)	AI	1 MΩ in parallel with 100 pF[1] or 10 kΩ in parallel with 40 pF[2]	—	—	—	—	±300 pA

Table 1. I/O Terminal Summary (Continued)

Terminal Name	Terminal Type and Direction	Impedance Input/ Output	Protection (Volts) On/Off	Source (mA at V)	Sink (mA at V)	Rise Time (ns)	Bias
AO 0	AO	50 Ω	Short-circuit to ground	5 at 10	5 at –10	—	—
AO 1	AO	50 Ω	Short-circuit to ground	5 at 10	5 at –10	—	—
AI GND	—	—	—	—	—	—	—
D GND	—	—	—	—	—	—	—
+5 V	—	0.1 Ω	Short-circuit to ground	1 A	—	—	—
P0.<0..7>	DIO	—	V_{CC} + 0.5	13 at (V_{CC} – 0.4)	24 at 0.4	1.1	50 kΩ pu
EXTSTROBE*	DO	—	—	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 0/AI START TRIG	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 1/ AI REF TRIG	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 2	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 3/ CTR 1 SOURCE	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 4/ CTR 1 GATE	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
CTR 1 OUT	DO	—	—	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 5/ AO SAMP CLK	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 6/ AO START TRIG	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 7/ AI SAMP CLK	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 8/ CTR 0 SOURCE	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
PFI 9/ CTR 0 GATE	DIO	—	V_{CC} + 0.5	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu

Table 1. I/O Terminal Summary (Continued)

Terminal Name	Terminal Type and Direction	Impedance Input/ Output	Protection (Volts) On/Off	Source (mA at V)	Sink (mA at V)	Rise Time (ns)	Bias
CTR 0 OUT	DO	—	—	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu
FREQ OUT	DO	—	—	3.5 at (V_{CC} – 0.4)	5 at 0.4	1.5	50 kΩ pu

* Indicates active low

[1] Applies to range ≤ ±10 V, impedance refers to AI <0..3>–.

[2] Applies to range > ±10 V, impedance refers to AI <0..3>–.

AI = Analog Input DIO = Digital Input/Output pu = pull-up
AO = Analog Output DO = Digital Output

Note: The tolerance on the 50 kΩ pull-up resistors is large. Actual value might range between 17 kΩ and 100 kΩ.

Specifications

The following specifications are typical at 25 °C unless otherwise noted.

Analog Input

Input Characteristics

Number of channels .... 4 pseudodifferential

Type of ADC

Resolution

NI 6115 .... 12 bits, 1 in 4,096

NI 6120 .... 16 bits, 1 in 65,536

Pipeline

NI 6115 4

NI 6120 0

Sampling rate

Maximum

NI 6115 .... 10 million S/s

NI 6120 .... 800 kS/s

Minimum

NI 6115 .... 20 kS/s

NI 6120 .... No minimum

Input impedance

AI + to AI –

Range ≤ ±10 V .... 1 MΩ in parallel with 100 pF

Range > ±10 V .... 10 kΩ in parallel with 40 pF

AI – to AI GND

NI 6115 .... 100 GΩ in parallel with 10 nF

NI 6120 .... 100 GΩ in parallel with 100 pF

AI + to AI GND

NI 6115 .... 100 GΩ in parallel with 100 pF

NI 6120 .... 100 GΩ in parallel with 100 pF

Input bias current .... ±300 pA

Input offset current .... ±200 pA

Input coupling .... DC/AC

Max working voltage for all analog input channels

Positive input (AI +) .... ±42 V for ±20 V and ±42 V ranges; ±11 V for other ranges

Negative input (AI –) .... ±2.5 V

Overvoltage protection (AI +, AI –) .... ±42 V

Input current during overvoltage conditions .... ±20 mA max

Input FIFO size .... 16 or 32 MS

Data transfers .... DMA, interrupts, programmed I/O

DMA mode .... Scatter-gather

DC Transfer Characteristics

INL

NI 6115 .... ±0.35 LSB typ, ±1 LSB max

NI 6120 .... ±2.5 LSB max

DNL

NI 6115 .... ±0.25 LSB typ, ±1 LSB max

NI 6120 .... 0.75 LSB typ, no missing codes

Offset, gain error .... Refer to Tables 2 and 3[1]

Table 2. NI 6115 Analog Input DC Accuracy Information

Nominal Range (V)	Absolute Accuracy							Relative Accuracy	
	% of Reading			Noise + Quantization (mV)				Resolution (mV)	
Full Scale	24 Hours	1 Year	Offset[1] (mV)	Single Pt.	Averaged	Temp Drift (%/°C)	Absolute Accuracy at Full Scale (±mV)	Single Pt.	Averaged
±42	0.346	0.348	33	42	3.6	0.0229	211.0	48	4.8
±20	0.271	0.273	13	17	1.4	0.0229	69.4	19	1.9
±10	0.026	0.028	6.7	8.3	0.72	0.0004	10.22	10	1.0
±5	0.016	0.018	3.4	4.2	0.36	0.0004	4.61	4.8	0.48
±2	0.036	0.038	1.3	1.8	0.16	0.0004	2.26	2.1	0.21
±1	0.043	0.045	0.68	1.1	0.09	0.0004	1.23	1.2	0.12
±0.5	0.058	0.060	0.35	0.69	0.061	0.0004	0.71	0.80	0.080
±0.2	0.103	0.105	0.15	0.43	0.039	0.0004	0.39	0.51	0.051

[1] The offset might degrade by 2.25 LSB with filter enabled.

Table 3. NI 6120 Analog Input DC Accuracy Information

Nominal Range (V)	Absolute Accuracy							Relative Accuracy	
	% of Reading			Noise + Quantization (μV)				Resolution (μV)	
Full Scale	24 Hours	1 Year	Offset[1] (μV)	Single Pt.	Averaged	Temp Drift (%/°C)	Absolute Accuracy at Full Scale (±mV)	Single Pt.	Averaged
±42	0.1571	0.1588	8,356.7	6,094.8	549.3	0.0106	87.4766	7,232.7	723.3
±20	0.1388	0.1405	3,343.7	2,437.9	219.7	0.0106	31.3295	2,893.1	289.3
±10	0.0327	0.0344	1,672.6	1,219.0	109.9	0.0006	5.0552	1,446.5	144.7
±5	0.0354	0.0371	837.1	750.1	68.7	0.0006	2.6777	904.1	90.4
±2	0.0393	0.0409	366.3	337.3	31.1	0.0006	1.1825	409.9	41.0
±1	0.0771	0.0788	275.5	196.4	18.3	0.0006	1.0652	241.1	24.1
±0.5	0.1000	0.1017	184.3	107.5	10.1	0.0006	0.6946	132.6	13.3
±0.2	0.1229	0.1246	93.0	54.0	5.1	0.0006	0.3440	67.5	6.8

[1] The offset might degrade by 8 LSB with filter enabled and by 1 LSB when sampling above 500 kS/s.

[1] Accuracies are valid for measurements following an internal calibration. Averaged numbers assume dithering and averaging of 100 single-channel readings. Measurement accuracies are listed for operational temperatures within ±1 °C of internal calibration temperature and ±10 °C of external or factory-calibration temperature.

Dynamic Characteristics

Analog filters

Number

NI 6115....................................2

NI 6120....................................1

Type

NI 6115....................................3-pole Bessel

NI 6120....................................5-pole Bessel

Frequency

NI 6115....................................50 and 500 kHz (software-enabled)

NI 6120....................................100 kHz (software-enabled)

Crosstalk ....–80 dB, DC to 100 kHz

Table 4. NI 6115 Analog Input Dynamic Characteristics

Input Range	Bandwidth[1] (MHz)	SFDR Typ[2] (dB)	CMRR[3] (dB)	System Noise[4] (LSB_{rms})
±42 V	5.5	78	34	0.35
±20 V	4.4	78	40	0.45
±10 V	7.2	81	46	0.35
±5 V	4.8	81	52	0.35
±2 V	4.8	85	60	0.45
±1 V	4.4	85	66	0.60
±500 mV	4.4	85	70	0.80
±200 mV	4.1	81	72	1.3

[1] –3 dB frequency for input amplitude at 96% of the input range (–0.3 dB)
[2] Measured at 100 kHz with twelfth-order bandpass filter after signal source
[3] DC to 60 Hz
[4] LSB_{rms}, not including quantization

Table 5. NI 6120 Analog Input Dynamic Characteristics

Input Range	Bandwidth[1] (MHz)	SFDR Typ[2] (dB)	CMRR[3] (dB)	System Noise[4] (LSB_{rms})
±42 V	1.0	95	60	1.2
±20 V	1.0	96	68	1.2
±10 V	1.0	95	76	1.2
±5 V	1.0	95	82	1.5
±2 V	1.0	96	90	1.7
±1 V	1.0	94	95	2.0
±500 mV	1.0	90	100	2.2
±200 mV	1.0	85	105	2.8

[1] –3 dB frequency for input amplitude at 10% of the input range (–20 dB)
[2] Measured at 100 kHz with twelfth-order bandpass filter after signal source
[3] DC to 60 Hz
[4] LSB_{rms}, not including quantization

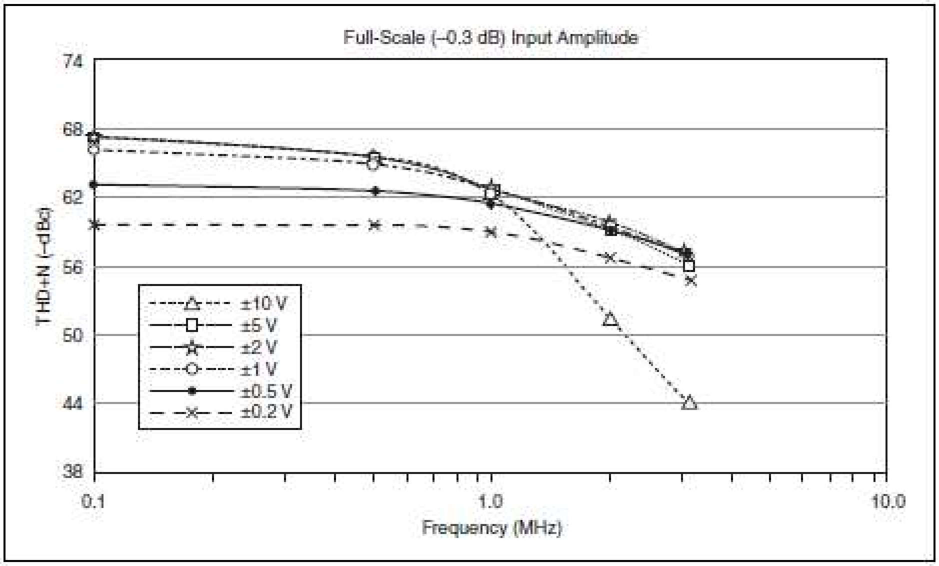


Figure 1. NI 6115 Total Harmonic Distortion Plus Noise (THD+N)

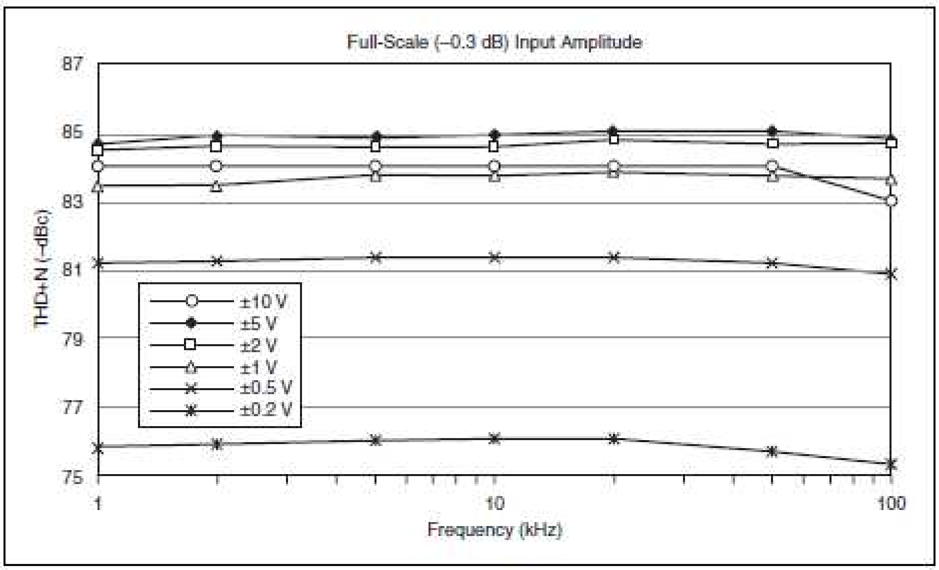


Figure 2. NI 6120 Total Harmonic Distortion Plus Noise (THD+N)

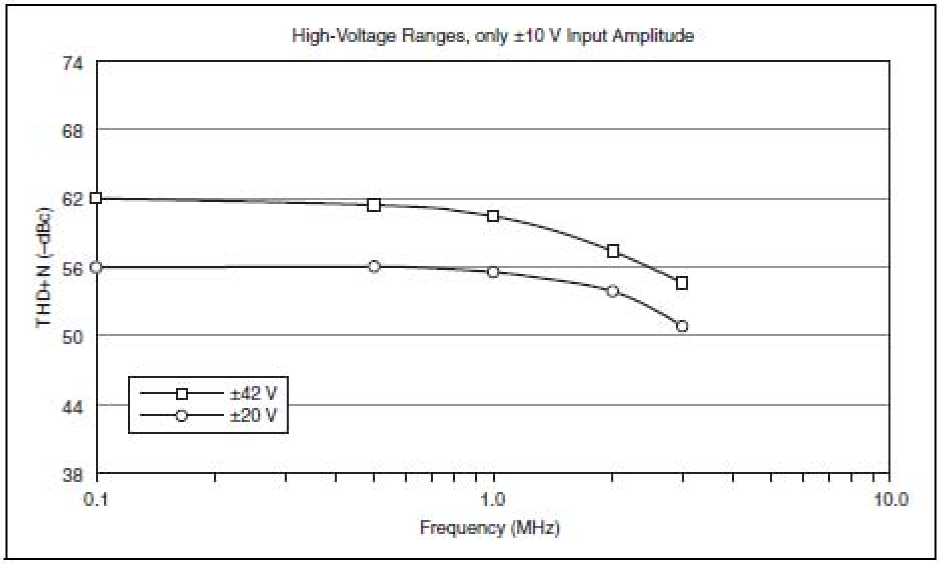


Figure 3. NI 6115 High-Voltage THD+N

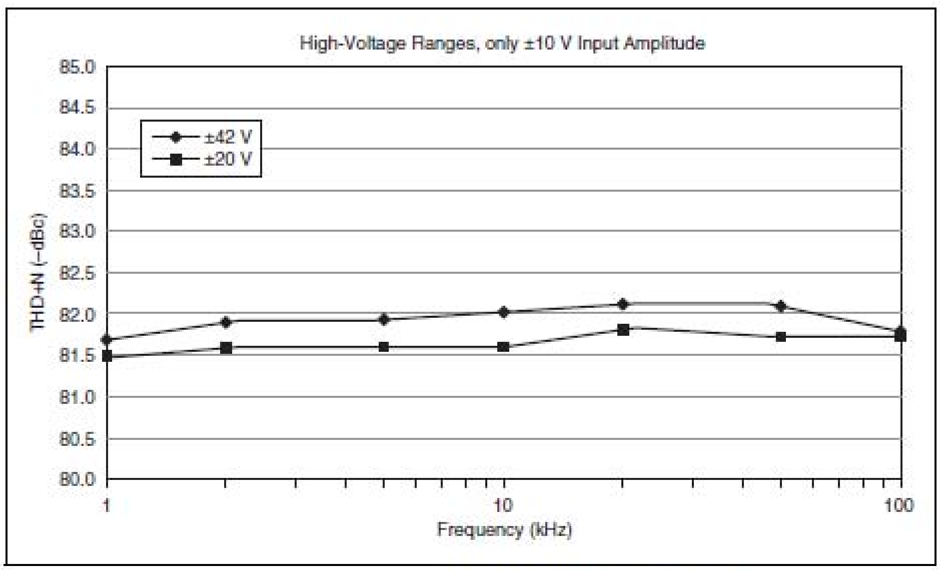


Figure 4. NI 6120 High-Voltage THD+N

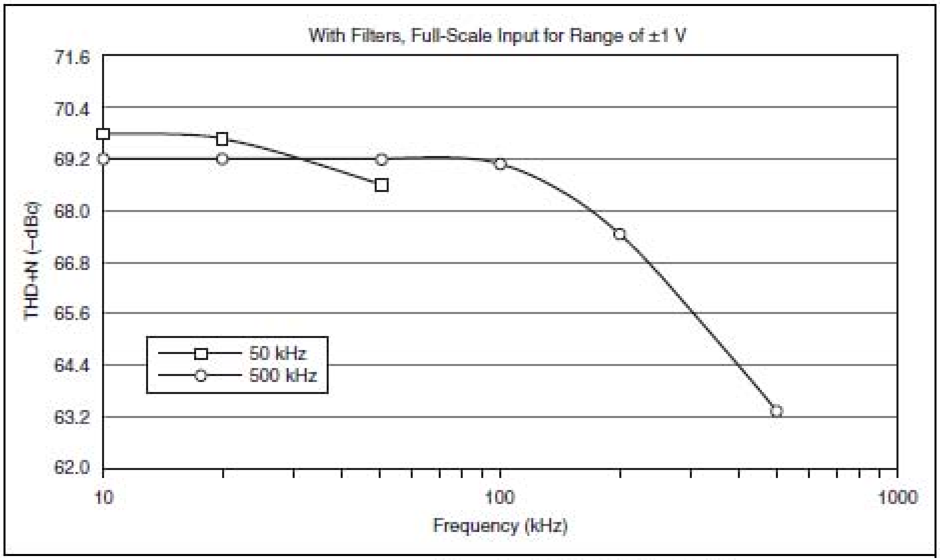


Figure 5. NI 6115 THD+N with Filters

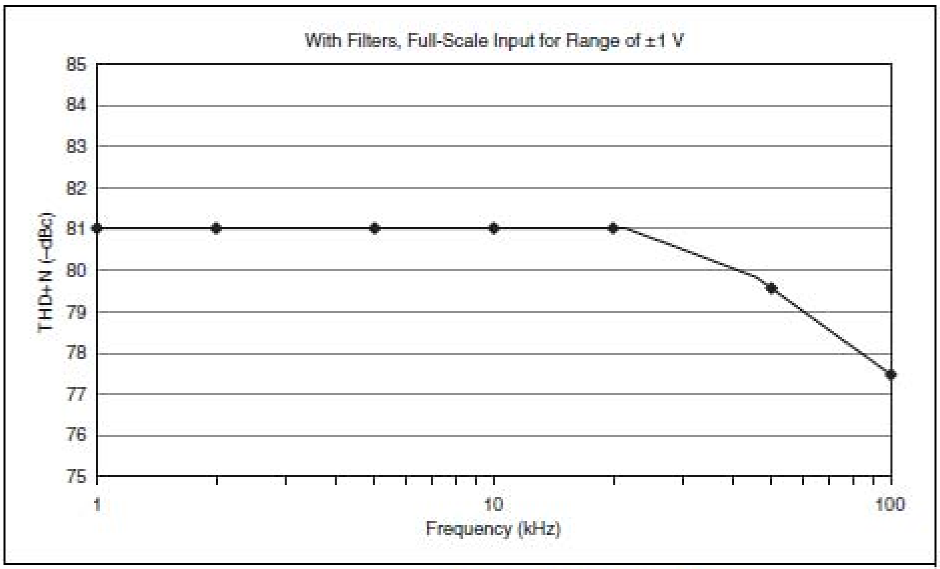


Figure 6. NI 6120 THD+N with Filters

Stability

Recommended warm-up time .... 15 min

Calibration interval .... 1 year

Offset temperature coefficient
- Pregain
  - NI 6115 .... ±12 µV/°C
  - NI 6120 .... ±1.5 µV/°C
- Postgain
  - NI 6115 .... ±64 µV/°C
  - NI 6120 .... ±2.1 LSB/°C

Gain temperature coefficient
- NI 6115 .... ±21.3 ppm/°C
- NI 6120 .... ±22.2 ppm/°C

Analog Output

Output Characteristics

Number of channels .... 2 voltage

Resolution
- NI 6115 .... 12 bits, 1 in 4,096
- NI 6120 .... 16 bits, 1 in 65,536

Max update rate
- 1 channel .... 4 MS/s, system dependent
- 2 channel .... 2.5 MS/s, system dependent

Output buffer size.................................. 16 or 32 MS

Data transfers .... DMA, interrupts, programmed I/O

DMA modes... Scatter-gather

DC Transfer Characteristics

INL
- NI 6115 .... ±0.5 LSB typ, ±2 LSB max
- NI 6120 .... ±0.35 LSB typ, ±1 LSB max

DNL

NI 6115 .... ±0.25 LSB typ, ±1 LSB max
NI 6120 .... ±0.2 LSB typ, ±1 LSB max

Offset, gain error

NI 6115 .... Refer to Table 6
NI 6120 .... Refer to Table 7

Table 6. NI 6115 Analog Output DC Accuracy Information

Nominal Range at Full Scale (V)	Absolute Accuracy						Relative Accuracy
	% of Reading			Offset (mV)	Temp Drift (%/°C)	Absolute Acc. at Full Scale (mV)	Theoretical Resolution (mV)
	24 Hours	90 Days	1 Year				
±10	0.0437	0.0445	0.0454	8.9	0.0006	13.5	4.88

Table 7. NI 6120 Analog Output DC Accuracy Information

Nominal Range at Full Scale (V)	Absolute Accuracy						Relative Accuracy
	% of Reading			Offset (μV)	Temp Drift (%/°C)	Absolute Acc. at Full Scale (mV)	Theoretical Resolution (μV)
	24 Hours	90 Days	1 Year				
±10	0.0511	0.0519	0.0528	1,864.4	0.0006	6.7	305.2

Voltage Output

Ranges .... ±10 V

Output coupling .... DC

Output impedance .... 50 Ω ±5%

Current drive .... ±5 mA min

Output stability .... Any passive load

Protection .... Short-circuit to ground

Power-on output voltage
(before software loads calibration values)

NI 6115 .... ±400 mV
NI 6120 .... ±80 mV

Initial power-up glitch

Magnitude .... ±2 V
Duration .... 200 ms

Dynamic Characteristics

Slew rate

NI 6115 .... 300 V/µs
NI 6120 .... 15 V/µs

Noise

NI 6115 .... 600 μV_{rms}, DC to 5 MHz
NI 6120 .... 100 μV_{rms}, DC to 1 MHz

Glitch energy at midscale transition

NI 6115 .... ±30 mV for 1 µs
NI 6120 .... ±10 mV for 1 µs

Settling time

NI 6115 .... 300 ns to 0.01%
NI 6120 .... 4 µs to ±1 LSB

Stability

Offset temperature coefficient

NI 6115 .... ±35 µV/°C
NI 6120 .... ±35 µV/°C

Gain temperature coefficient

NI 6115 .... ±56.9 ppm/°C
NI 6120 .... ±6.5 ppm/°C

Calibration

Level.. 5.000 V (±2.5 mV)
(actual value stored in EEPROM)

Temperature coefficient .... ±2.0 ppm/°C max

Long-term stability................................ $\pm 6\ \text{ppm}/\sqrt{1{,}000\ \text{h}}$

Digital I/O

Number of channels .... 8 input/output

Compatibility .... TTL/CMOS

Table 8. Digital Logic Levels

Level	Min	Max
Input low voltage	0.0 V	0.8 V
Input high voltage	2.0 V	5.0 V
Input low current (V_{in} = 0 V)	—	–320 µA
Input high current (V_{in} = 5 V)	—	10 µA
Output low voltage (I_{OL} = 24 mA)	—	0.4 V
Output high voltage (I_{OH} = –13 mA)	4.35 V	—

Power-on state ....Input (high-impedance)

Data transfers..DMA, interrupts, programmed I/O

Input buffer...2,048 bytes

Output buffer ....2,048 bytes

Transfer rate (1 word = 8 bits)................10 Mwords/s

Timing I/O

Number of channels.................................2 up/down counter/timers, 1 frequency scaler

Resolution

- Counter/timers ....24 bits
- Frequency scaler..............................4 bits

Compatibility ....TTL/CMOS

Base clocks available

- Counter/timers ....20 MHz, 100 kHz
- Frequency scaler..............................10 MHz, 100 kHz

Base clock accuracy...............................±0.01%

Max source frequency............................20 MHz

Min source pulse duration ....10 ns, edge-detect mode

Min gate pulse duration ....10 ns, edge-detect mode

Data transfers .... DMA, interrupts, programmed I/O

DMA modes... Scatter-gather

Triggers

Analog Trigger

Source... All analog input channels, external trigger (PFI 0/AI START TRIG)

Level

- Internal .... ± full-scale
- External... ±10 V

Slope... Positive or negative (software-selectable)

Resolution

- NI 6115 .... 8 bits, 1 in 256
- NI 6120 .... 12 bits, 1 in 4,096

Hysteresis .... Programmable

Bandwidth (–3 dB) .... 5 MHz internal/external

External input (PFI 0/AI START TRIG)

- Impedance....................................... 10 kΩ
- Coupling.. AC/DC
- Protection .... –0.5 V to (V_{CC} + 0.5) V when configured as a digital signal, ±35 V when configured as an analog trigger signal or disabled, ±35 V powered off

Digital Trigger

Compatibility .... TTL

Response .... Rising or falling edge

Pulse width.. 10 ns min

RTSI Trigger Lines (PCI Only)

Trigger lines <0..6> 7

RTSI clock 1

PXI Trigger Bus (PXI Only)

Trigger lines <0..6> 7

Star trigger 1

Bus Interface

Type ....Master, slave

Power Requirement

+5 VDC (±5%)

NI 6115 ....2.2 A

NI 6120 ....3.0 A

+3.3 V ....0.8 A

Power available at I/O connector............+4.65 to +5.25 VDC at 1 A

Physical

Dimensions (not including connectors)

NI PCI-6115/6120 ....31.2 by 10.6 cm (12.3 by 4.2 in.)

NI PXI-6115/6120...........................16 by 10 cm (6.3 by 3.9 in.)

I/O connector ....68-pin male SCSI-II type

Maximum Working Voltage

Maximum working voltage refers to the signal voltage plus the common-mode voltage.

Channel-to-earth ....42 V, Measurement Category I

Channel-to-channel.................................42 V, Measurement Category I

Environmental

Operating temperature............................ 0 to 50 °C

Storage temperature .... –20 to 70 °C

Humidity .... 10 to 90% RH, noncondensing

Maximum altitude .... 2,000 m

Pollution Degree (indoor use only)........ 2

Safety

The NI 6115/6120 is designed to meet the requirements of the following standards of safety for electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN 61010-1
- UL 61010-1
- CAN/CSA-C22.2 No. 61010-1

Note For UL and other safety certifications, refer to the product label, or visit `ni.com/certification`, search by model number or product line, and click the appropriate link in the Certification column.

Electromagnetic Compatibility

Emissions .... EN 55011 Class A at 10 m
FCC Part 15A above 1 GHz

Immunity .... EN 61326:1997 + A2:2001,
Table 1

EMC/EMI.. CE, C-Tick, and FCC Part 15
(Class A) Compliant

Note For EMC compliance, operate this device with shielded cabling.

CE Compliance

This product meets the essential requirements of applicable European Directives, as amended for CE marking, as follows:

Low-Voltage Directive (safety) 73/23/EEC

Electromagnetic Compatibility
Directive (EMC) 89/336/EEC

Note Refer to the Declaration of Conformity (DoC) for this product for any additional regulatory compliance information. To obtain the DoC for this product, visit `ni.com/certification`, search by model number or product line, and click the appropriate link in the Certification column.

Source Code

The following is the listing of the different MATLAB source code modules used to process the data acquired the time division multiplexed multi-beam LDV system. If this source code is cut and pasted directly into a MATLAB file for execution, be aware that several of the lines have wrapped and the pasted source code should be checked to make sure that the formatting is correct.

Listing for TDMVelChan

```
function vc = TDMVelChan(ch)
% Function to calculate the time division multiplexed velocity for one
% channel from the 16 beam MB LDV for a patch on the ground
% as it move at a constant speed.
% There are issues with the size of the data, 128 megasamples per channel.
% We will not only have to process each channel by itself, but we will have
% to process each channel in manageable blocks

% Carrier Clock Frequency
clk = 10*10^6; % 10 MHz AOM frequency

% block size
blkTime = 0.06;  % Time in seconds. This is the size of data chunk that we
                 % can handle easily. If we need finer control, use smaller
                 % number.

% Precessing Flags - Set as needed
% Quick look time
qlt = 20/200;  % The test souce is at 200 Hz, this will let us look at 100
               % cycles of the test signal
quicklook = 0; % Set this flag to do a quick look based on above time

% Channel Start
chanStart = 0;  % If chanStart is >0 then the program will start reading
                % data at beginning of that channel

% Track the carrier frequecy shift due to Doppler shift from the speed of
% the vehicle
trackf = 2; % set to zero to use a fixed carrier frequency
            % set to 1 to track the carrier frequency for each channel
            % set to 2 to track the carrier frequency for each process
            %    block

%Check for file name
if nargin == 1
    [fn, fp] = uigetfile('*.*', 'TDM-LDV data file');
    if ((length(fn)==1) && (length(fp)==1) && (fn == 0) && (fp == 0))
        error('File not selected by user');
    end
    cd(fp);
    [fp fn]
else
    error('need channel number to process');
end

% open the file and retrieve the header info
fid = fopen(fn, 'r');
if fid > 0
    ver = fread(fid,1,'int32','b');
    switch ver
```

```
        case 2050
            % This is specific to the Signetic data since the data is
            % stored channel concatenated, not channel interleaved.
            nc = fread(fid,1,'int32','b'); % number of channels
%            rl = fread(fid,1,'float64','b'); % record length (seconds)
            sr = fread(fid,1,'int32','b'); % sample rate
            scale = fread(fid,1,'float64','b'); % Volts per bit
            sc = fread(fid,1,'int32','b');   % samples per channel
            SpdPtr = fread(fid, 1, 'int64', 'b'); % Pointer to start of speed data
            spr = fread(fid, 1, 'float64', 'b'); % spatial resolution cm

        otherwise
            fclose(fid);
            error('Data reader does not support this version of data - Try MBVread');

    end
else
    error('Error - could not open %s \n', fn);
end
hdr = ftell(fid);
rl = sc/sr;

% find real samples per channel
dataptr = sc*2*nc+hdr; % where the data should end
datadiff = (SpdPtr-dataptr)/2; % Convert pointer difference into sample difference
if datadiff ~= 0
    % Adjust sc count
    sc = sc + datadiff/nc;
end
clear dataptr datadiff

% Open the output file
ofn = [strtok(fn, '.') '.vel'];
[ofn, opn] = uiputfile('*.vel', 'Velocity Output File', ofn);
fido = fopen(ofn, 'w');
if fido <=0
    error('Error - could not open %s \n', [opn '\' ofn]);
end

% Load filter constants
% bandpass filter the raw photodector signal around the 10 MHz
% for 50 Mfs
load 50Mfs_5_15Mbpf_fir45.mat;
% The next two filters were for testing the noise associated with
% wide band filters and the need for carrier tracking
%load 50Mfs_9980_10020Kbpf_fir45.mat;
%load 50Mfs_9936_10064Kbpf_fir45.mat;
bb = Num;
ab = 1;
dpl = length(bb); % determine pad length. This is used to determine the
                % length of the pad data that is put before and after
                % actual data. The maximum length is a function of the
                % number of filter sections, so this will be set to be
                % the length of the longest filter section.

% Filter after mixing - remove frequencies > 1/2 the carrier frequency
load 100Mfs_5Mlpf_fir45.mat;
bm = Num;
am = 1;
plt = length(bm); % determine pad length
if plt > dpl
    dpl = plt;
end
```

```
% Realize that decimation could be included after the mixing since the
% filter could be set to be much lower. This would reduce the processing
% load down stream since the size of the data vector would be smaller and
% the number of filter sections would be less.

% Decimation
% 80 MHz sample rate to 2.5 KHz velocity sample rate
% 80/.0025 = 32,000 -> we will do this in two stages
% Stage 1
decim1 = 50; % 100 MHz -> 2 MHz sample rate
% LPF will have an Fc of 800 KHz
load 100Mfs_800Klpf_fir45;
b1 = Num;  %Filter Coefficients for stage 1 decimation after demodulation
a1 = 1;
plt = length(b1); % Determine pad length
if plt > dpl
    dpl = plt;
end

% Stage 2
decim2 = 50; % 2 MHz -> 40 KHz sample rate
% LPF will have an Fc of 16 KHz
load 2Mfs_16Klpf_fir45;
b2 = Num; %Filter Coefficient for stage 2 decimation after demodulation
a2 = 1;
plt = length(b2); % Determine pad length
if plt > dpl
    dpl = plt;
end

% Stage 3
decim3 = 16; % 40 KHz -> 2.5 KHz sample rate
% LPF will have an Fc of 1 KHz
load 40Kfs_1Klpf_fir45;
b3 = Num;
a3 = 1;
plt = length(b3); % Determine pad length
if plt > dpl
    dpl = plt;
end

% Calculate file read block size
rblksz = sr*blkTime;
nblk = floor(rl/blkTime + 1/sr); % make sure this is an integer
                                 % and that it doesn't round down.

% See if we have enough extra to padd the processing block - this is
% differtn from the pl that is defined above. This provides enough pad to
% take care of severe ringing that takes place during the initial band pass
% filtering as well as takes care of the reduction in the data vector due
% to taking the derivative during demodulation. Therefore we will still
% have exactly 0.06 seconds of data to filter and save after demodulation.
if (nblk*rblksz+dpl > sc)
    error('Can not pad data for processing');
end

rsc = sc;  %raw samples / channel;

% Adjusted sample rate & samples per channel (50 -> 100 MHz)
sr = sr*2;
sc = sc*2;
```

```
% Calculate process block size
pblksz = sr*blkTime;

% Calculate the clock time series
dt = 1/sr;
t=0:dt:((pblksz+2*(2*dpl))/sr)-dt; %for upsampled data
if trackf == 0
    iclock = single(sin(2*pi*clk*t)); %Make the clock
    qclock = single(sin(2*pi*clk*t-pi/2));
    clear t;
else
    df = 1/blkTime;
    fd = 0:df:sr/4-df;
end

if quicklook == 1
    pblk = floor(qlt/blkTime); % find out how many blocks we really need to process
    if pblk < 1
        pblk = 1;
    end
    % refigure all of header info
    rl =pblk*blkTime;
    sc = rl*sr;
else
    pblk = nblk;
end

% Update sample rate to reflect decimation
decim = decim1*decim2*decim3;
srd = sr/decim;
scd = nblk*pblksz/decim;

% Output the header for the output velocity file
fwrite(fido, ver+1, 'int32', 'b'); % The version number modified so that this
                                   % program can't read it by mistake
if ch == 0;
    fwrite(fido, nc, 'int32', 'b'); % The number of channels
else
    fwrite(fido, 1, 'int32', 'b'); % We are only doing 1 channel
end
rlPtrLoc = ftell(fido); % for later update
fwrite(fido, rl, 'float64', 'b'); % run length in seconds
fwrite(fido, srd, 'float64', 'b'); % The velocity sample rate
fwrite(fido, scale, 'float64', 'b'); % The scale factor volts/bit
scPtrLoc = ftell(fido); %for later update
fwrite(fido, scd, 'int32', 'b'); % samples per channel
fidoSpdPtrLoc = ftell(fido); % Get the location of the SpdPtr header location
fwrite(fido, 0, 'int64', 'b'); % Speed data location pointer
fwrite(fido, spr, 'float64', 'b'); % Spatial resolution (cm)
% fwrite(fido, ve, 'int32', 'b'); % Number of velocity elements
% fwrite(fido, vi, 'int32', 'b'); % velocity elements
% fwrite(fido, vsr, 'float64', 'b'); % velocity sample rate

% Need to consider that I may either need to pad or window the data before
% processing.

vc.v = zeros(nc, scd, 'single');

hWait = waitbar(0,'Please wait... Processing');

if ch == 0 % Treat it like we are going to do each channel
    waitMax = nc*pblk;
    for cnt = 1:nc
```

```
    % Cycle through each channel getting the data and doing the demod
    % for looking at a short segment of the data "quick look"
    if quicklook == 1
        doffset = hdr + rsc*2*(cnt-1);
        fseek(fid, doffset, 'bof');
    end
    if chanStart > 0
        doffset = hdr + rsc*2*(chanStart-1);
        fseek(fid, doffset, 'bof');
        cnt = chanStart;
        chanStart = 0; %Only do this once
    end
    vcnt = 0;
    for blk = 1:pblk
        waitValue = blk + (cnt-1)*pblk;
        waitbar(waitValue/waitMax, hWait);
         if blk == 1 % Figure out the data pad
            [data ns] = fread(fid, rblksz+dpl, 'int16', 'b');
            data = [fliplr(data(1:dpl)') data']; % Add the pad to the beginning
            fidx = ftell(fid); %get the current index
            fidx = fidx - dpl * 2; % Adjust the file pointer for the next
            fseek(fid, fidx, 'bof'); % read, taking into account the overlap
            ns = ns + dpl;
        else
            [data ns] = fread(fid, rblksz+2*dpl, 'int16', 'b');
            fidx = ftell(fid);
            fidx = fidx - (2*dpl) * 2;
            fseek(fid, fidx, 'bof');
            data = data';
        end
        if ns ~= rblksz+2*dpl
            warning('Expected %d data elements, read %d data elements', rblksz,
ns);
        end
        % process the data
        dm = mean(data);
        data = data - dm;

        if ((trackf == 2) || (trackf == 1 && blk == 1))
            D = abs(fft(data(dpl+1:end-dpl)));
            o = findPeak(fd,D(1:end/2));
            carrierF = o(1,1);
            if trackf == 1
                vc.cf(cnt) = carrierF;
            end
            if trackf == 2
                vc.cf(blk,cnt) = carrierF;
            end
            if ((carrierF < 5e6) || (carrierF > 15e6))
                iclock = zeros(1,length(t), 'single');
                qclock = iclock;
            else
                iclock = single(sin(2*pi*carrierF*t)); %Make the clock
                qclock = single(sin(2*pi*carrierF*t-pi/2));
            end
            clear D;
        end

        % Bandpass before upsampling
        data = filtfilt(bb, ab, data);

        % Upsample the data from 50 to 100 M samples / sec
        data = single(interp(data, 2));
```

```
    % Create the baseband I & Q
    [ibase, qbase] = IQ(iclock, qclock, data, am, bm);

    % demodulate the I & Q
    v = IQdemodv(ibase, qbase, dt); % velocity
    % Remove the data pad
    v = v(2*dpl:end-(2*dpl-1));

    % Stage 1
    % filter the velocity signal
    % Pad
    pl = length(b1);
    vp = [v(1:pl), v, v(end-pl+1:end)];
    vp = filtfilt(b1, a1, vp);
    v = vp(pl+101:end-(pl-100));

    % decimate the velocity signal
    v = downsample(v, decim1, floor(decim1/2));

    % Stage 2
    % filter the velocity signal
    % Pad
    vp = [v(1:pl), v, v(end-pl+1:end)];
    vp = filtfilt(b2, a2, vp);
    v = vp(pl+1:end-pl);

    % decimate the velocity signal
    v = downsample(v, decim2, floor(decim2/2));

    %Stage 3
    % filter the velocity signal
    % Pad
    pl = length(b3);
    vp = [v(1:pl), v, v(end-pl+1:end)];
    vp = filtfilt(b3, a3, vp);
    v = vp(pl+1:end-pl);

    % decimate the velocity signal
    v = downsample(v, decim3, floor(decim3/2));

    % match end points
    if blk == 1
        epv = v(end);
    else
        bias = v(1)-epv; % find out what the bias is
        v = v - bias;
        epv = v(end); % get the new enpoint value
    end

    % Write the data to file
    fwrite(fido, v, 'float64', 'b'); % Velocity data

    % Track number of velocity points written
    vc.v(cnt,vcnt+1:vcnt+length(v)) = v;
    vcnt = vcnt + length(v);

end
if vcnt ~= scd
    warning('Samples per channel not as expected')
    warning('Expected %d wrote %d samples for channel %d', scd, vcnt, cnt)
end
cpt = ftell(fid);
```

```
        edp = rsc*2*cnt+hdr; % Where the file pointer should be after reading
                             % the all the data from the channel
    %testing
    fprintf(1,'Channel: %2d, file pointer difference: %d\n', cnt, edp-cpt);

        if cpt < edp
            fseek(fid,(edp - cpt), 'cof');
        end
    end %End of indexing through each channel

else % we are only doing one channel (note this could be changed to allow
                                    % vector of channels)
    % Find the index into the data
    waitMax = pblk;
    cnt = ch;
    doffset = hdr + rsc * 2 * (cnt-1);
    fseek(fid, doffset, 'bof');
    vcnt = 0;
    for blk = 1:pblk
        waitValue = blk;
        waitbar(waitValue/waitMax, hWait);
        if blk == 1 % Figure out the data pad
            [data ns] = fread(fid, rblksz+dpl, 'int16', 'b');
            data = [fliplr(data(1:dpl)') data']; % Add the pad to the beginning
            fidx = ftell(fid); %get the current index
            fidx = fidx - dpl * 2; % Adjust the file pointer for the next
            fseek(fid, fidx, 'bof'); % read, taking into account the overlap
            ns = ns + dpl;
        else
            [data ns] = fread(fid, rblksz+2*dpl, 'int16', 'b');
            fidx = ftell(fid);
            fidx = fidx - (2*dpl) * 2;
            fseek(fid, fidx, 'bof');
            data = data';
        end
        if ns ~= rblksz+2*dpl
            warning('Expected %d data elements, read %d data elements', rblksz, ns);
        end
        % process the data
        dm = mean(data);
        data = data - dm;

        if ((trackf == 2) || (trackf == 1 && blk == 1))
            D = abs(fft(data(dpl+1:end-dpl)));
            o = findPeak(fd,D(1:end/2));
            carrierF = o(1,1);
            if trackf == 1
                vc.cf(cnt) = carrierF;
            end
            if trackf == 2
                vc.cf(blk,cnt) = carrierF;
            end
            if ((carrierF < 5e6) || (carrierF > 15e6))
                iclock = zeros(1,length(t), 'single');
                qclock = iclock;
            else
                iclock = single(sin(2*pi*carrierF*t)); %Make the clock
                qclock = single(sin(2*pi*carrierF*t-pi/2));
            end
            clear D;
        end

        % Bandpass before upsampling
```

```
    data = filtfilt(bb, ab, data);

    % Upsample the data from 50 to 100 M samples / sec
    data = single(interp(data, 2));

    % Create the baseband I & Q
    [ibase, qbase] = IQ(iclock, qclock, data, am, bm);

    % demodulate the I & Q
    v = IQdemodv(ibase, qbase, dt); % velocity
    % Remove the data pad
    v = v(2*dpl:end-(2*dpl-1));

    % Stage 1
    % filter the velocity signal
    % Pad
    pl = length(b1);
    vp = [v(1:pl), v, v(end-pl+1:end)];
    vp = filtfilt(b1, a1, vp);
    v = vp(pl+101:end-(pl-100));

    % decimate the velocity signal
    v = downsample(v, decim1, floor(decim1/2));

    % Stage 2
    % filter the velocity signal
    % Pad
    vp = [v(1:pl), v, v(end-pl+1:end)];
    vp = filtfilt(b2, a2, vp);
    v = vp(pl+1:end-pl);

    % decimate the velocity signal
    v = downsample(v, decim2, floor(decim2/2));

    %Stage 3
    % filter the velocity signal
    % Pad
    pl = length(b3);
    vp = [v(1:pl), v, v(end-pl+1:end)];
    vp = filtfilt(b3, a3, vp);
    v = vp(pl+1:end-pl);

    % decimate the velocity signal
    v = downsample(v, decim3, floor(decim3/2));

    % match end points
    if blk == 1
        epv = v(end);
    else
        bias = v(1)-epv; % find out what the bias is
        v = v - bias;
        epv = v(end); % get the new enpoint value
    end

    % Write the data to file
    fwrite(fido, v, 'float64', 'b'); % Velocity data

    % Track number of velocity points written
    vc.v(cnt,vcnt+1:vcnt+length(v)) = v;
    vcnt = vcnt + length(v);

end
if vcnt ~= scd
```

```
        warning('Samples per channel not as expected')
        warning('Expected %d wrote %d samples for channel %d', scd, vcnt, cnt)
    end
    cpt = ftell(fid);
    edp = rsc*2*cnt+hdr; % Where the file pointer should be after reading
                         % the all the data from the channel

%testing - Comment out the following line
fprintf(1,'Channel: %2d, file pointer difference: %d\n', cnt, edp-cpt);

    if cpt < edp
        fseek(fid,(edp - cpt), 'cof');
    end
end
close(hWait);

% Clear out unused variables
clear a* b* Num data ibase iclock qbase qclock;

% % Make sure we get to the speed data
% ptr = ftell(fid);
% if (ptr ~= eofptr - (sdsz+shoffset))
%     warning(' Adjusting file pointer to access speed data');
%     fseek(fid, -(sdsz+shoffset), 'eof')
% end

% Put the speed data location in the header of the output file
fidoSpdPtr = ftell(fido);
fseek(fido, fidoSpdPtrLoc, 'bof');
fwrite(fido, fidoSpdPtr, 'int64', 'b'); % We know where to find the speed data

% Update rl to reflect the actual number of samples processes
if rl ~= scd/srd
    warning('run length %d does not agree with sc/sr %d - adjusting', rl, scd/srd)
    rl = scd/srd;
    fseek(fido, rlPtrLoc, 'bof');
    fwrite(fido, rl, 'float64', 'b'); % run length in seconds
end

fseek(fido, fidoSpdPtr, 'bof'); % get back to where we write speed hdr/data

% Get the velocity data
fseek(fid, SpdPtr, 'bof'); % Put the file pointer where it needs to be
ssrl = fread(fid, 1, 'float64', 'b'); % speed signal record length (seconds)
fwrite(fido, ssrl, 'float64', 'b');
sssr = fread(fid, 1, 'int32', 'b'); % speed sample rate
fwrite(fido, sssr, 'int32', 'b');
ssdatacnt = fread(fid, 1, 'int32', 'b'); % Number of speed sensor data points
fwrite(fido, ssdatacnt, 'int32', 'b');
[spd num] = fread(fid, ssdatacnt, 'float64', 'b');
if num ~= ssdatacnt
    warning('Expected %d data elements in speed signal, read %d data elements',
ssdatacnt, num);
end
fwrite(fido, spd, 'float64', 'b');

fclose(fid);
fclose(fido);

return

function [ib, qb] = IQ(iclk, qclk, sig, a, b)
    ifi=iclk.*sig;
```

```
    ifq=qclk.*sig;
    % Filter the I & Q
    ib = filtfilt(b, a, ifi);
    qb = filtfilt(b, a, ifq);
return

function vel = IQdemodv(ib, qb, dt)
% Wavelength of laser light
lamda = 532e-9;
outlier = 1; % Set to 0 (zero) to not fix large velocity spikes

N=length(ib);
k=2:N-1; %index array for the derivative calculation
idot=(ib(k+1)-ib(k-1))/(2*dt); %Time correlated derivitive to signal
qdot=(qb(k+1)-qb(k-1))/(2*dt);
num = ib(k).*qdot - idot.*qb(k);
den = ib(k).^2 + qb(k).^2;
den = ZeroCk(den);
vel = num./den;
if outlier==1
    vMean = mean(abs(vel));
    thresh = 10*vMean;
    vIndx = abs(vel)>(thresh);
    vSign = sign(vel(vIndx));
    vel(vIndx) = vSign*thresh; %Replace the outliers with the threshold value
end
vel = (lamda/(4*pi))*vel; % scale to m/s
return

function d = ZeroCk(d)
dzero=find(d==0);
ldzero = length(dzero);
lden = length(d);
if ldzero~=0;
    for j=1:ldzero
        dindex = dzero(j);
        if j==1     %Check if we're at the very first element of the array
            if dindex==1    %If so we can only take the next element, not the average
                d(1)=d(2);
            end
            if d(1) == 0; % Make sure we don't have two at the beginning
                d(1) = 1e-6;
            end
        elseif j==ldzero    %Check if we're at the very last element of the array
            if dindex==lden %If so we can only take the previous element, not the
average
                d(end) = d(end-1);
            end
            if d(end) == 0; % Make sure that we don't have two at the end
                d(end) = 1e-6;
            end
        else
            d(dindex)=(d(dindex-1)+d(dindex+1))/2;
            if(d(dindex)==0) % Final bit of paranoia
                d(dindex) = 1e-6;
            end
        end
    end
end
return
```

Appendix D

Listing for TDMdmux

```
function o = TDMdmux(fn)
% Function to retrieve the speed data from the raw data file and make the
% index vector for demultiplexing the velocity data
BeamGeo = 1; % 0 -> Don't correct for beam geometry
             % 1 -> Correct for beam geometry
Offset = 0;  % 0 -> Don't apply an offset
             % 1 -> Use an offset
Plotflg = 0; % 0 -> Don't do plots per channel
             % 1 -> Do plots per channel

%Check for file name
if nargin == 0
    [fn, fp] = uigetfile('*.vel', 'TDM-LDV data file');
    if ((length(fn)==1) && (length(fp)==1) && (fn == 0) && (fp == 0))
        error('File not selected by user');
    end
    cd(fp);
    [fp fn]
end

% open the file and retrieve the header info
fid = fopen(fn, 'r');
if fid > 0
    ver = fread(fid,1,'int32','b');
    switch ver
        case 2051 % We will use the raw data for now, but this will eventually
                  % transfer over to the processed velocity file
            % TDM-LDV data since the data had been demodulate and stored
            % stored channel concatenated, not channel interleaved.
            nc = fread(fid, 1, 'int32', 'b'); % number of channels
            rl = fread(fid, 1, 'float64', 'b'); % record length (seconds)
            sr = fread(fid, 1, 'float64', 'b'); % sample rate
            scale = fread(fid, 1, 'float64', 'b'); % Volts per bit
            sc = fread(fid,1, 'int32', 'b');   % samples per channel
            SpdPtr = fread(fid, 1, 'int64', 'b'); %Pointer to speed data
            spr = fread(fid, 1, 'float64', 'b'); % spatial resolution cm

        otherwise
            fclose(fid);
            error('Data reader does not support this version of data - Try MBVread');

    end
else
    error('Error - could not open %s \n', fn);
end
if rl ~= sc/sr
    warning('run length %d does not agree with sc/sr %d - adjusting', rl, sc/sr)
    rl = sc/sr;
end;
dt = 1/sr; % define the time between samples
t = 0:dt:rl-dt; % define the time axis, useful for plots
hdr = ftell(fid); % A pointer to the end of the velocity header data and
                  % begining of the velocity data

% Read speed data
fseek(fid, SpdPtr, 'bof');
ssrl = fread(fid,1,'float64','b'); % speed signal record length (seconds)
sssr = fread(fid,1,'int32','b'); % speed signal sampling rate
ssdatacnt = fread(fid,1,'int32','b'); % speed signal data count
[ssdata  num]= fread(fid,ssdatacnt,'float64','b'); % data from the photocell
```

```
if num ~= ssdatacnt
    warning('Expected %d data elements in speed signal, read %d data elements', ...
        ssdatacnt, num);
end

% put pointer back to where it should be
fseek(fid,hdr,'bof'); % Get ready to read velocity data

% read the velocity data
[data num] = fread(fid, nc*sc, 'float64', 'b');
if num ~= nc*sc
    warning ('Expectd %d data elements in velocity data, read %d data elements', ...
        nc*sc, num);
end
data = reshape(data, sc, [])';

% Bias the speed data so that there are zero crossings
bias = .1*(max(ssdata)-min(ssdata));
ssdata = (ssdata - bias)';

% Find Positive zero crossings
zcp = zeroCross(ssdata, 'p');

% Do some preliminary data validation
% Make sure some body didn't back up the cart while speed data was being
% taken
deltazcp = zcp(2:end) - zcp(1:end-1);
endrun = find(deltazcp>10*mean(deltazcp));
if length(endrun) > 0
    zcp = zcp(1:endrun); % This gets rid of extra data at the end of a run
end

% Check to see if we ran off the speed track over a marker (remove the
% extra dip withing the marker pulse)
deltazcp = zcp(2:end) - zcp(1:end-1);
indx = deltazcp>mean(deltazcp)/2;
if(indx(end) == 1)
    indx = [indx true];
end
zcp = zcp(indx);

% Remove offset
zco = zcp(1);
zcp = zcp - zco;

% Check the speed pulses to see if we have a valid set
% check one - see if we have more than 1 meter worth of pulses
if length(zcp)*spr < 1
    % we have scanned less than a meter
    warning('Scanned less than one meter: %d scanned', length(zcp)*spr);
end

% check two - see if the time difference between pulses is greater than
% twice the average
zdiff = zcp(2:end)-zcp(1:end-1);
zmean = 2*mean(zdiff);
zmax = max(zdiff);
if zmax >= zmean
    zidx = find(2*zdiff>zmean);
    warning('Have a speed pulse dt > 2*mean %d > %d at idx %d',zmax, zmean, zidx(1));
end

% Average Speed
```

```
speed = spr./(zdiff/sssr);
aspeed = mean(speed);


% Calculate demux index
%  First positive zero cross will correspond to the trigger, so this is
%  when data acquisition started.
% Determine offset scale factor from speed index using ratio of sample
% rates
srRatio = sr/sssr;

if BeamGeo == 1
% Correct for the beam geometry
% 1.46 degrees per beam - included angle of 22 degrees
% beam 1 is 22 degrees from NADar
% beam 16 is at NADAR
% spacing in cm - same units as spr
% ** Note that we are estimating the distance that beam one moves (1st entry)
% as the system moves forward based on the distance between beam 16 and 15.
    beamCorrect = [6.9 6.8 6.7 6.6 6.5 6.35 6.3 6.25 6.15 6.05 6.0 6.0 6.0 6.0 5.95
5.95];
else
    beamCorrect = 6*ones(1,16);
end
% Based on beam spacing, determine the number of velocity markers that are
% within the data
midx  = rl*sssr; % Find the max possible index
dmidx = find(zcp>midx);
if isempty(dmidx)
    zcpd = zcp(1:end-1);
else
    zcpd = zcp(1:dmidx(1)-1);
end

% Take the first cut at getting the indexing
velIdx = (zcpd * srRatio);

% Determine the number of patches based on the average beam spacing
avgbs = sum(beamCorrect)/length(beamCorrect);
numPatch = round((length(velIdx))*spr/avgbs) - nc;

lineVel = zeros(numPatch, sc);
% Rearange the velocity data based on the index
idxCnt = 1;
if Offset == 0
    offset = 0;
    gndOffset = 0;
else
    offset = 69;
    gndOffset = 0;
end
maxVelLen = 0; % Maximum velocity line length
firstTime = true;
for pn = 0:numPatch
    vel =[];
    if ((gndOffset > zcp(idxCnt)) || (firstTime == true))
        firstTime = false;
        idxCnt = idxCnt + 1;
        nspe = zcp(idxCnt + 1) - zcp(idxCnt);
        ispeed = spr/(nspe/sssr); % Short time average speed
        gndOffsetInc = floor(sssr*beamCorrect(end)/ispeed);
        offsetInc = floor(sr*beamCorrect(end)/ispeed);
%         Idx = round([0 sr*beamCorrect/ispeed]);
```

```
        Idx = round([0 srRatio*nspe*beamCorrect/spr]);
        vIndx = zeros(1,nc+1);

        for bn = 2:nc+1
            vIndx(bn) = sum(Idx(1:bn));
        end
    end

    for sn = 1:nc % Cycle through the segments
        b1 = vIndx(sn)+1;
        b2 = vIndx(sn+1);
        IDX(pn+1,sn,:) = ([b1 b2]+offset);
        vel = [vel data(sn, offset+b1:offset+b2)];
    end
    offset = offset+offsetInc;
    gndOffset = gndOffset + gndOffsetInc;

%     for bn = 1:nc
%         Idx = floor(velIdx*beamCorrect(bn)/spr) + 1; % remember: Matlab's index
starts at 1
%         b = Idx(bn+pn);
%         s = Idx(bn+pn+1)-1;
%         vel = [vel data(bn,b:s)];
%     end
    velLen = length(vel);
    if velLen > maxVelLen
        maxVelLen = velLen; % capture the longest velocity segment
    end
    lineVel(pn+1,1:velLen) = vel;
    if (offset+vIndx(end)>length(data));
        break;
    end
end

if Plotflg ==1
    for i = 1:nc
        figure;
        plot(data(i,:));
        grid on
        vline(IDX(:,i,1),'g');
    end
end

% Truncate the lineVel array to maxVelLen
lineVel = lineVel(1:pn+1,1:maxVelLen);

% Open the output file
ofn = [strtok(fn, '.') '.vdm']; % vdm extension to denote demultiplexed velocity data
[ofn, opn] = uiputfile('*.vdm', 'Demultiplexed Velocity Output File', ofn);
fido = fopen(ofn, 'w');
if fido <=0
    error('Error - could not open %s \n', [opn '\' ofn]);
end

% Output the header for the output velocity file
fwrite(fido, ver+1, 'int32', 'b'); % The version number modified so that this
                                   % program can't read it by mistake
fwrite(fido, numPatch, 'int32', 'b'); % The number of patches
fwrite(fido, velLen*sr, 'float64', 'b'); % run length in seconds
fwrite(fido, sr, 'float64', 'b'); % The velocity sample rate
fwrite(fido, scale, 'float64', 'b'); % The scale factor volts/bit
fwrite(fido, length(speed), 'int32', 'b'); % the number of speed elements
fwrite(fido, speed, 'float64', 'b'); % Speed elements
```

```
fwrite(fido, floor(rl*aspeed), 'int32', 'b'); % Estimated distance scanned (cm)
fwrite(fido, spr, 'float64', 'b'); % Spatial resolution (cm)
%fwrite(fido, ve, 'int32', 'b'); % Number of velocity elements
fwrite(fido, sssr, 'int32', 'b'); % velocity sample rate
fwrite(fido, zcp, 'int32', 'b'); % Postive zero crossing events vector (length ve)
% fwrite(fido, zcn, 'int32', 'b'); % Negative zero crossing events vector (length ve)
fwrite(fido, lineVel', 'float64', 'b'); % The line velocity data
fclose(fid);
fclose(fido);
o.vel = lineVel;
o.speed = speed;
o.zcp = zcp;
o.data = data;
o.IDX = IDX;
```

Appendix D

Listing for Find Peak

```
function [o,idx]=findPeak(x,y)
% Function to find the positive peak (local maximums) locations of a vector
% Y using the first dirivative and zero crossings
% X is the x axis values associated with y
if nargin == 1;
     y = x;
     x = 1:length(y);
end

d = fderiv(y); % find first derivitive.
zc = zeroCross(d,'n','th'); % find the zero crossings for positive peaks
zc = zc+1; % Account for the fact that the derivative is one count off.
           % Now zc should correspond to indices that point to the
           % local peak(s) in Y

% Noticed that sometimes, the real peak(s) is/are further along the vector
% therefore check the next few samples to see if they are bigger. This
% usually only requires one itteration through the while loop.
if length(y)==zc(end) % watch out for maximum at the very end
    p=find(y(zc(1:end-1))<y(zc(1:end-1)+1));
    while ~isempty(p)
        zc(p) = zc(p)+1;
        p=find(y(zc(1:end-1))<y(zc(1:end-1)+1));
    end
else
    t1 = y(zc);
    t2 = y(zc+1);
    p=find(t1<t2);
    while ~isempty(p)
        zc(p) = zc(p)+1;
        t1 = y(zc);
        t2 = y(zc+1);
        p=find(t1<t2);
    end
end

% Check to see if there is a peak at the first value, if so find out where
% it goes in the array
p = find(y(zc) < y(1));
if ~isempty(p); % The initial value is greater than some of the found peaks
    if(zc(1) == 2); % Then this peak is actually at the first index
        zc(1) = 1;
    else % Add the first value to the beginning of zc
        zc = [1 zc];
    end
end

% Now order the peaks from largest to smallest
pky = y(zc);
[PKY,PI] = sort(pky,'descend');
pkx = x(zc);
PKX = pkx(PI);

o(1,:) = PKX;
o(2,:) = PKY;

if nargout == 2
    idx = zc(PI);
end
```

Appendix D

Listing for fderiv

```
function [op, o]=fderiv(x);
% function to find the numerical deriviative of x using mid point
% derivative technique
% Y(n+1)-Y(n-1)/2 yields the slope based on the points either side of Y(n)


c=x(3:end);
op=(c-x(1:end-2))/2;

% Check the number of output arguments. If it is 2 then return the right
% length of the original time series so that the original signal and the
% derivative are the same length.

if nargout == 2
    o = x(2:end-1);
end

%b=x(2:end);
%o=b-x(1:end-1);
```

Listing for Zero Cross

```
function o=zeroCross(x,t,f)
% This function finds all of the zero crossings of a function and returns
% the index of the location.
% Zero crossings are found for three different types
%  b - bipolar zero crossing - both positive and negative zero crossings
%  p - positive going zero crossing - positive going zero crossing
%  n - negative going zero crossing - negative going zero crossing
% f -> denotes type of filter
%  th -> threshold based on a ratio between peak and rms value
%  sb -> Single bit fluctuations remove two adjacent zero crossings
%  bo -> do both threshold and single bit fluctuations
%  anything else, do nothing

if nargin < 2 % Check to see if we need to set the default zerocross type
    t='b';  % find both positive and negative zero crossings
    f = ''; % No filter is the default
end
if nargin < 3
    f = ''; % No filter is the default
end

if (strcmp(f, 'th') || strcmp(f, 'bo'))
    pk = max(abs(x));
    xrms = norm(x)/sqrt(length(x));
    threshold = pk/(10*(pk/xrms));
    threshold = .2;
    idx = find((x<threshold)&(x>0));
    x(idx) = 0;
    idx = find((x>-threshold)&(x<0));
    x(idx) = 0;
end

b = x >= 0; % convert array into logicals
sc = b(2:end)-b(1:end-1); % Compare with its neighbor. If it is the same
                          % result will be 0, if it is different then it
                          % will be either + or - 1, which means that we
                          % had a sign change or zero crossing
if(strcmp(f, 'sb') || strcmp(f, 'bo'))
% Remove single bit fluctuations
    r = find(sc~=0);
    rd = r(2:end)-r(1:end-1);
    rdd = find(rd == 1); % The difference in index is 1, so there are two
                     % adjacent sign changes
    b(r(rdd)+1) = b(r(rdd));
    sc = b(2:end) - b(1:end-1); % Redo with filtered data
end

o =[];
if (any(t=='n') || any(t =='b'));
    o = find(sc<0);
end
if (any(t=='p') || any(t=='b'));
    p = find(sc>0);
    o = [o p+1];
    if any(t=='b')
        o = sort(o);
    end
end
```

Filter Coefficients

100Mfs_5_15Mbpf_fir45 Filter Coefficients

Generated by MATLAB(R) 7.8 and the Signal Processing Toolbox 6.11.
Generated on: 22-Jun-2009 15:55:24
Coefficient Format: Decimal
Discrete-Time FIR Filter (real)
Filter Structure : Direct-Form FIR
Filter Length : 125
Stable : Yes
Linear Phase : Yes (Type 1)

Numerator:

```
 0.0012748491573441538
-0.0026166073730901577
-0.0013992382650939111
-0.00054878107460993024
-0.000053806734239070632
-0.00039445022988362718
-0.0015524890133719071
-0.0026356516547932302
-0.0025212551521786638
-0.00095546530320111381
 0.00099769012341845132
 0.0017887262557221928
 0.00084989466718591184
-0.00068082441826700047
-0.00087068813365974364
 0.001150538422388991
 0.0041323108686404106
 0.0056351060798890953
 0.0042935371515483928
 0.0013021332543417246
-0.00048406815999349945
 0.00075826301398836766
 0.0038196740550252382
 0.0052669329344453876
 0.0026167749528278779
-0.0030043594469143921
-0.0074666317250314685
-0.0073166086726376978
-0.0032498635575440942
 0.00013093112844140325
-0.0015120072344954702
-0.0078143648559564367
-0.013408945613329839
-0.012572263749240722
-0.0048243878854721091
 0.0039640355836718308
 0.006600699454531362
 0.00151072070167523
-0.0050605929345356329
-0.0043507145561072534
 0.0064136244814296441
 0.020322628605512903
 0.026178493231379898
 0.018854522872793836
 0.0050570378926388113
-0.0018072376413204511
 0.0058840916105876934
 0.021393164421892284
 0.027966108597045014
 0.013818162962422047
-0.015262619269792164
-0.038403768019309802
-0.037392611302774395
-0.015197337133554407
 0.0024466021356977737
-0.012768978197629803
-0.064224674964302342
-0.11930288986495545
-0.12778209089145856
-0.059382795850203146
 0.06761775373879296
 0.1908931276698406
 0.241833657438358
 0.1908931276698406
 0.06761775373879296
-0.059382795850203146
-0.12778209089145856
-0.11930288986495545
-0.064224674964302342
-0.012768978197629803
 0.0024466021356977737
-0.015197337133554407
-0.037392611302774395
-0.038403768019309802
-0.015262619269792164
 0.013818162962422047
 0.027966108597045014
 0.021393164421892284
 0.0058840916105876934
-0.0018072376413204511
 0.0050570378926388113
 0.018854522872793836
 0.026178493231379898
 0.020322628605512903
 0.0064136244814296441
-0.0043507145561072534
-0.0050605929345356329
 0.00151072070167523
 0.006600699454531362
 0.0039640355836718308
-0.0048243878854721091
-0.012572263749240722
-0.013408945613329839
-0.0078143648559564367
-0.0015120072344954702
 0.00013093112844140325
-0.0032498635575440942
-0.0073166086726376978
-0.0074666317250314685
-0.0030043594469143921
 0.0026167749528278779
 0.0052669329344453876
 0.0038196740550252382
 0.00075826301398836766
-0.00048406815999349945
 0.0013021332543417246
 0.0042935371515483928
 0.0056351060798890953
 0.0041323108686404106
 0.001150538422388991
-0.00087068813365974364
-0.00068082441826700047
 0.00084989466718591184
 0.0017887262557221928
 0.00099769012341845132
-0.00095546530320111381
-0.0025212551521786638
-0.0026356516547932302
-0.0015524890133719071
-0.00039445022988362718
-0.000053806734239070632
-0.00054878107460993024
-0.0013992382650939111
-0.0026166073730901577
 0.0012748491573441538
```

100Mfs_800Klpf_fir45 Filter Coefficients

Generated by MATLAB(R) 7.8 and the Signal Processing Toolbox 6.11.
Generated on: 22-Jun-2009 15:58:44
Coefficient Format: Decimal
Discrete-Time FIR Filter (real)
Filter Structure : Direct-Form FIR
Filter Length : 1234
Stable : Yes
Linear Phase : Yes (Type 2)

Numerator:

```
-0.0028198205780140565
 0.000076395283552097536
 0.000076661984535180382
 0.000075508158206514468
 0.000076366567932588132
 0.00007577032092086558
 0.00007717836162892712
 0.000077085729392808045
 0.000078997189411837698
 0.000079345887525347249
 0.000081709756873558307
 0.000082428641002914652
 0.000085195261840568907
 0.000086197584639391249
 0.000089325779675198443
 0.000090499805870497465
 0.000093974108573662354
 0.00009513236986817596
 0.00009905842015937253
 0.000099416577350158741
 0.00010565468409777165
 0.00010663680022482414
 0.0001071759235033139
 0.0001101646465786546
 0.00011292724537424661
 0.00011549092437890653
 0.00011827917561999532
 0.00012034812217346291
 0.00012284850378707461
 0.00012442771496268355
 0.00012654926738739011
 0.00012764690899380897
 0.00012933937322024769
 0.0001299515207842006
 0.00013116893748943773
 0.00013127836665178865
 0.00013196984735976093
 0.00013155945975404718
 0.00013165286858759411
 0.00013076832806300587
 0.00013027801303914458
 0.0001279264948065204
 0.00012793574700008335
 0.00012540150196365838
 0.0001236783306207581
 0.00012041879593506933
 0.00011802771549413414
 0.00011423751416412371
 0.00011124376365828029
 0.00010686291664000142
 0.00010320746336116684
 0.000098173655417675438
 0.000093825457333526589
 0.000088126871241741515
 0.000083097395446525644
 0.000076767944085206
 0.00007110013761986618
 0.000064201533892440082
 0.000057951318511700618
 0.000050582272923321934
 0.000043687605142000464
 0.000036210450885439723
 0.00002886904426262259
 0.000020447135008449992
 0.000013169759800744398
 0.0000047232471735315223
-0.0000028403160805324547
-0.000011536458197057044
-0.000019329086894088066
-0.000028138140048195297
-0.000035981083891447208
-0.000044742098498718689
-0.00005249286518665356
-0.000061092219468918303
-0.000068645140897852404
-0.000076993588255282186
-0.000084253300525510139
-0.000092260090378026936
-0.000099118811303544781
-0.00010670007151901287
-0.00011300444746429206
-0.00012003916396033289
-0.00012593738245925959
-0.00013196846615529329
-0.00013712775306687353
-0.00014263929775265756
-0.0001469677393142084
-0.00015155154684283901
-0.00015489303178660039
-0.00015849098742972681
-0.0001608411489000673
-0.00016343457873385864
-0.00016475565789722708
-0.0001662812980484478
-0.00016650515906696832
-0.0001668956216989495
-0.00016596415684399764
-0.00016517719358254352
-0.00016306276417511626
-0.00016108318302178806
-0.00015777182536080327
-0.00015467113078508708
-0.00015004481167425884
-0.00014583198275159833
-0.00014023369232932844
-0.00013471346760123622
-0.00012793971408940705
-0.00012135786818006939
-0.0001135737597760581
-0.00010597953615275367
-0.000097193718518075525
-0.000088612753633763534
-0.000078889124084297914
-0.000069423455994695009
-0.000058886689202743234
-0.000048666047744277876
-0.000037450431234486992
-0.000026601762226956694
-0.000014834724958041542
-0.0000034803948071975011
 0.0000086902876342089853
 0.000020444325312044768
 0.000032867650288383263
 0.000044749488283311097
 0.000057445101527889113
 0.000069337865002288158
 0.000081881745329259042
 0.000093660906896834723
 0.00010602380766471312
 0.00011753731515115953
 0.00012951010341717174
 0.00014054096516596557
 0.00015193707916485479
 0.00016232659440101275
 0.00017300604705350374
 0.000182610514700927
 0.00019242422508625959
 0.00020108542395707446
 0.00020987034837344773
```

0.00021742478629184362	-0.00036498634989638604	0.00048882244495281708
0.00022503037344462698	-0.0003672050119213374	0.00047818242697997542
0.00023133621975764953	-0.00036781358900952674	0.00046544168582642476
0.00023758886451544139	-0.00036763905727959676	0.0004514066315465747
0.00024260421142313247	-0.00036576311987010978	0.00043534102299676476
0.00024737284335290969	-0.00036315422265930253	0.00041801755730318853
0.00025084267103707078	-0.00035882381207530661	0.00039872917268365905
0.00025416987591782465	-0.00035371384274510967	0.00037827068111251621
0.0002560887810210662	-0.00034687885169742538	0.00035595388932737866
0.00025777535551880929	-0.00033928679188414751	0.00033256838667891598
0.00025801757303272428	-0.00033000377113186338	0.00030743712827310333
0.00025801994090392157	-0.00031998872232773734	0.00028135355724374271
0.00025657491656392185	-0.00030831295608087835	0.00025366149574817089
0.00025487186623017596	-0.00029593552328226755	0.00022516829469897484
0.00025170386981805074	-0.0002819486752807473	0.00019522728378608759
0.00024825316677369938	-0.00026732115050411747	0.00016465250338366976
0.00024333464088812772	-0.00025115486712243111	0.00013280490625410543
0.00023813352544945323	-0.00023442666264361558	0.00010050481289764672
0.00023148137846121632	-0.00021624181816448521	0.000067118977783893041
0.0002245683064246687	-0.00019758605232928547	0.000033475843757728007
0.00021623251675856542	-0.00017755838795338206	-0.0000010267482754306584
0.00020766819328766652	-0.00015717767311484608	-0.000035610778606344484
0.00019769923111891781	-0.00013553943008691319	-0.000070807489690653033
0.0001875804189429048	-0.00011362489670508789	-0.00010585526734065761
0.0001760454230776724	-0.000090653734473817704	-0.00014132351978431807
0.00016439927787793591	-0.000067509213900208195	-0.00017640582433084764
0.00015151427514765309	-0.000043412155480999221	-0.00021166052750568737
0.00013845596299060653	-0.000019313533840499119	-0.00024629707240219527
0.000124219166580545	0.0000055721892928241659	-0.0002808786205883095
0.00010993965360518287	0.000030317385361566199	-0.0003146096584042469
0.000094580277856859841	0.000055706747009405706	-0.00034804414435529512
0.000079243662698242311	0.000080802586229005487	-0.00038038916321840935
0.000062899756543591603	0.0001063719928901365	-0.00041219815242038698
0.000046660621387630604	0.00013147613766815655	-0.00044269464033252137
0.000029520662130661391	0.00015687654067901352	-0.00047243124842208687
0.000012593323469278252	0.00018164959434224673	-0.00050063490924587284
-0.0000051183448352880648	0.00020654724271713465	-0.00052786055412808348
-0.000022516456553598949	0.00023065388526395799	-0.00055333879128422348
-0.000040584325141533248	0.00025471111246339138	-0.00057763247123221841
-0.000058239219929641633	0.00027781006367451778	-0.0005999707441854598
-0.000076443984548822501	0.00030068328278034899	-0.00062095145155948101
-0.000094129058136657034	0.00032242749378139047	-0.00063978359363386006
-0.00011222576604668537	0.00034380458315989024	-0.00065707496656227608
-0.0001296961492806099	0.00036385424336953442	-0.00067209173575583005
-0.00014746101937459545	0.00038340241243765968	-0.00068538601691430162
-0.00016441022240285636	0.00040150460778300969	-0.00069625949971620013
-0.00018162289447892256	0.00041890012353078786	-0.00070529402306138654
-0.00019788626118805381	0.00043471981145224011	-0.00071180085541535168
-0.00021421312632923569	0.00044971140783373661	-0.0007163559127458057
-0.00022952522075358117	0.0004630009194159148	-0.00071829119544377078
-0.00024480524266682051	0.0004753189165866735	-0.00071819809114411848
-0.00025895826823021592	0.0004858145033054726	-0.00071543124093910926
-0.00027293984093171081	0.0004952273344279879	-0.00071059092190144807
-0.00028567932086350596	0.00050272864842492316	-0.00070304684983063742
-0.00029813612119934619	0.00050905874348756725	-0.00069340277944629491
-0.00030926083619526624	0.00051339669143979231	-0.00068105820536378198
-0.00032000760479768738	0.00051648051692205913	-0.00066662365005450011
-0.00032933008835131982	0.00051751170545672343	-0.00064952760312150457
-0.00033817688481614606	0.0005172270946876942	-0.00063038469798899115
-0.00034551404262245667	0.0005148548291701088	-0.0006086545449013347
-0.00035228747543676001	0.00051112816544740926	-0.00058494934563631728
-0.00035748270164706391	0.00050531036238359852	-0.00055874335165178365
-0.00036204393068751299	0.00049810541327826316	-0.00053069564455483862

-0.00050025083686440789	0.00028566617290912231	0.00033102966487666028
-0.00046810321671730522	0.0002234915111891775	0.00043198239322995904
-0.00043374133174993142	0.00015956522944163189	0.00053319669279704076
-0.00039783583101821347	0.000094523149082608674	0.00063404262179773572
-0.00035989449581343407	0.000028121666612714594	0.00073449672052161401
-0.00032060773623635583	-0.000038999752333575277	0.00083392434356337718
-0.00027950889528369736	-0.0001070705633499933	0.00093229600508296486
-0.00023728802129765201	-0.00017544612161619986	0.0010289748704705509
-0.0001934940361235705	-0.00024434363293221722	0.0011239203104572819
-0.00014882188419158707	-0.00031310817917260718	0.0012165099072220134
-0.00010283845934892658	-0.00038194519676740127	0.0013066965951852518
-0.000056251952617956524	-0.00045020665635144095	0.0013938701174600748
-0.0000086400182190129668	-0.00051808376411069607	0.0014779813065405115
0.000039289507771337641	-0.00058493577498299181	0.0015584389686721825
0.000087945383412639608	-0.00065093968380916566	0.0016351968150122133
0.00013662096835159558	-0.00071547103181023594	0.0017076733540648025
0.00018570946084584417	-0.00077868551512995984	0.0017758566019373228
0.00023451273096736528	-0.0008399787659662452	0.0018391632774067115
0.00028338725464581438	-0.00089951831183766526	0.0018976034746414901
0.00033167218432164462	-0.00095668104585849064	0.0019506292268018331
0.00037970072685910823	-0.0010116627618070345	0.0019982582438824851
0.00042679436899666599	-0.0010638538838922081	0.0020399698852100433
0.00047332593025963899	-0.0011134401405020092	0.0020758125538372502
0.00051860232221983106	-0.0011598302779329116	0.0021053012939125067
0.00056298040456686724	-0.0012032278838453804	0.0021285063696655062
0.00060578186324883754	-0.0012430616324811466	0.0021449802674732739
0.00064737426310801892	-0.0012795418118353229	0.0021548227178795369
0.00068708543961909507	-0.0013121193595638679	0.0021576220867702972
0.00072528071127283492	-0.0013410173964932773	0.0021535208769473055
0.00076129813782945584	-0.0013657127342492389	0.0021421442862359038
0.00079550675834895836	-0.0013864526074882015	0.002123668644488738
0.0008272641407302073	-0.0014027317246174749	0.0020977677228874157
0.00085694866961545831	-0.00141481951793788	0.0020646536032935953
0.00088392441013306023	-0.0014222377203500633	0.0020240509278105371
0.00090858280332597064	-0.0014252804551111897	0.0019762034734599168
0.00093029898870847097	-0.0014234973190842882	0.0019209020069706844
0.00094948306886115559	-0.0014172056781641736	0.0018584122758319708
0.00096551883122607523	-0.0014060013597168744	0.00178858019030063
0.00097884522055535945	-0.0013901968381628592	0.001711725998522884
0.00098886306265523775	-0.0013694453086006233	0.0016277302118003771
0.00099600124396552643	-0.0013440836910837044	0.0015369611061019496
0.00099972276851222151	-0.0013137815727844219	0.0014393570131921623
0.0010004436427659305	-0.0012789180941528548	0.0013353241058304483
0.0009976418991891659	-0.00123920249922804	0.0012248460292908921
0.00099177238342760187	-0.0011950339802983619	0.001108373499420467
0.00098233704063171697	-0.001146151770691882	0.00098594210839710429
0.00096979732956396563	-0.0010929871045531774	0.00085804132602843074
0.00095367757260468762	-0.0010353146297528912	0.00072476013817669195
0.00093446318622413807	-0.00097359278060084927	0.0005866221539868547
0.00091170537202771599	-0.00090763363368269109	0.00044375834241092843
0.00088590843103020494	-0.00083791359528353343	0.00029673866896819806
0.00085664890063087392	-0.00076428088185246084	0.00014573082758903653
0.00082444120729077046	-0.00068724218955202665	-0.0000086564015360848789
0.00078889517163202409	-0.00060667297521134825	-0.00016622086878493409
0.00075054317511755349	-0.00052310764577098495	-0.00032631461012709324
0.00070902141967763834	-0.00043644565550454836	-0.00048870637059933355
0.00066487861517135682	-0.00034725380307260544	-0.00065272219431096803
0.00061777520096100661	-0.00025544142663221298	-0.00081808161842529883
0.00056828027381864528	-0.00016161206312381715	-0.00098410743222504624
0.00051605786594639044	-0.000065702752502615723	-0.0011504811011481294
0.00046172646690290781	0.00003168649933427874	-0.0013165020184760152
0.0004049518838699479	0.00013057484784520989	-0.0014818362815834157
0.0003463592226354709	0.00023035256527318988	-0.0016457696457433247

-0.0018079424358925069	0.0075982462309454098	0.011953183474458288
-0.0019676314824506828	0.008046398944676925	0.011539889012966651
-0.0021244684162301363	0.008493889137702328	0.011119581573262106
-0.0022777242431054898	0.0089398689411820022	0.010692987558016571
-0.0024270242375341805	0.0093835861827764409	0.010260944031035897
-0.0025716450132131493	0.0098241950111563683	0.0098241950111563683
-0.0027111985825774912	0.010260944031035897	0.0093835861827764409
-0.0028449779801232996	0.010692987558016571	0.0089398689411820022
-0.0029725973714984756	0.011119581573262106	0.008493889137702328
-0.0030933621018964901	0.011539889012966651	0.008046398944676925
-0.0032068960700438291	0.011953183474458288	0.0075982462309454098
-0.0033125237060962903	0.012358636335892102	0.0071501596285552639
-0.003409889186666867	0.01275554538533619	0.0067029885555287194
-0.00349832747214672	0.013143114155810359	0.0062574391747704456
-0.0035775230908427959	0.013520652773112391	0.0058143447807188931
-0.0036468327778465997	0.013887407951264445	0.0053743900665045334
-0.0037059621870312613	0.014242718936035882	0.0049383867998628334
-0.0037543160508810725	0.014585868949721085	0.0045069811540848206
-0.0037916266700765309	0.014916236624631695	0.0040809603341974724
-0.003817335916435763	0.015233156610208881	0.0036609372966672667
-0.0038312189241855418	0.015536047481389542	0.0032476583127148788
-0.0038327668193246542	0.015824294243633744	0.0028416928943233071
-0.0038217931023824519	0.016097371092578854	0.0024437591969583847
-0.0037978395880742455	0.016354714760492963	0.0020543597625958679
-0.0037607694175273224	0.016595854196395494	0.0016741798841105676
-0.0037101727706956925	0.016820299830547555	0.0013036664877290158
-0.003645971616153181	0.017027627050223024	0.00094345352587472476
-0.0035678169838762605	0.017217420639752169	0.00059393104152329155
-0.00347567709518157	0.017389315574409342	0.0002556790912609506
-0.0033692729684057475	0.017542972127936346	-0.000070970359776215391
-0.003248624726074058	0.017678087036844384	-0.00038550428915550125
-0.0031135249699860701	0.017794398078673217	-0.00068764466145177774
-0.0029640445966621761	0.017891678630440701	-0.00097693916461130074
-0.0028000546804664754	0.017969729342138967	-0.0012531809713051878
-0.0026216838341772009	0.018028406005298959	-0.0015159742108734864
-0.0024288628261028575	0.018067586331780183	-0.0017651855228356887
-0.0022217988138884632	0.018087195053390105	-0.0020004832370599376
-0.0020004832370599376	0.018087195053390105	-0.0022217988138884632
-0.0017651855228356887	0.018067586331780183	-0.0024288628261028575
-0.0015159742108734864	0.018028406005298959	-0.0026216838341772009
-0.0012531809713051878	0.017969729342138967	-0.0028000546804664754
-0.00097693916461130074	0.017891678630440701	-0.0029640445966621761
-0.00068764466145177774	0.017794398078673217	-0.0031135249699860701
-0.00038550428915550125	0.017678087036844384	-0.003248624726074058
-0.000070970359776215391	0.017542972127936346	-0.0033692729684057475
0.0002556790912609506	0.017389315574409342	-0.00347567709518157
0.00059393104152329155	0.017217420639752169	-0.0035678169838762605
0.00094345352587472476	0.017027627050223024	-0.003645971616153181
0.0013036664877290158	0.016820299830547555	-0.0037101727706956925
0.0016741798841105676	0.016595854196395494	-0.0037607694175273224
0.0020543597625958679	0.016354714760492963	-0.0037978395880742455
0.0024437591969583847	0.016097371092578854	-0.0038217931023824519
0.0028416928943233071	0.015824294243633744	-0.0038327668193246542
0.0032476583127148788	0.015536047481389542	-0.0038312189241855418
0.0036609372966672667	0.015233156610208881	-0.003817335916435763
0.0040809603341974724	0.014916236624631695	-0.0037916266700765309
0.0045069811540848206	0.014585868949721085	-0.0037543160508810725
0.0049383867998628334	0.014242718936035882	-0.0037059621870312613
0.0053743900665045334	0.013887407951264445	-0.0036468327778465997
0.0058143447807188931	0.013520652773112391	-0.0035775230908427959
0.0062574391747704456	0.013143114155810359	-0.00349832747214672
0.0067029885555287194	0.01275554538533619	-0.003409889186666867
0.0071501596285552639	0.012358636335892102	-0.0033125237060962903

-0.0032068960700438291	0.0013066965951852518	-0.00038194519676740127
-0.0030933621018964901	0.0012165099072220134	-0.00031310817917260718
-0.0029725973714984756	0.0011239203104572819	-0.00024434363293221722
-0.0028449779801232996	0.0010289748704705509	-0.00017544612161619986
-0.0027111985825774912	0.00093229600508296486	-0.0001070705633499933
-0.0025716450132131493	0.00083392434356337718	-0.000038999752333575277
-0.0024270242375341805	0.00073449672052161401	0.00002812166612714594
-0.0022777242431054898	0.00063404262179773572	0.000094523149082608674
-0.0021244684162301363	0.00053319669279704076	0.00015956522944163189
-0.0019676314824506828	0.00043198239322995904	0.0002234915111891775
-0.0018079424358925069	0.00033102966487666028	0.00028566617290912231
-0.0016457696457433247	0.00023035256527318988	0.0003463592226354709
-0.0014818362815834157	0.00013057484784520989	0.0004049518838699479
-0.0013165020184760152	0.00003168649933427874	0.00046172646690290781
-0.0011504811011481294	-0.000065702752502615723	0.00051605786594639044
-0.00098410743222504624	-0.00016161206312381715	0.00056828027381864528
-0.00081808161842529883	-0.00025544142663221298	0.00061777520096100661
-0.00065272219431096803	-0.00034725380307260544	0.00066487861517135682
-0.00048870637059933355	-0.00043644565550454836	0.00070902141967763834
-0.00032631461012709324	-0.00052310764577098495	0.00075054317511755349
-0.00016622086878493409	-0.00060667297521134825	0.00078889517163202409
-0.0000086564015360848789	-0.00068724218955202665	0.00082444120729077046
0.00014573082758903653	-0.00076428088185246084	0.00085664890063087392
0.00029673866896819806	-0.00083791359528353343	0.00088590843103020494
0.00044375834241092843	-0.00090763363368269109	0.00091170537202771599
0.0005866221539868547	-0.00097359278060084927	0.00093446318622413807
0.00072476013817669195	-0.0010353146297528912	0.00095367757260468762
0.00085804132602843074	-0.0010929871045531774	0.00096979732956396563
0.00098594210839710429	-0.001146151770691882	0.00098233704063171697
0.001108373499420467	-0.0011950339802983619	0.00099177238342760187
0.0012248460292908921	-0.00123920249922804	0.0009976418991891659
0.0013353241058304483	-0.0012789180941528548	0.0010004436427659305
0.0014393570131921623	-0.0013137815727844219	0.00099972276851222151
0.0015369611061019496	-0.0013440836910837044	0.00099600124396552643
0.0016277302118003771	-0.0013694453086006233	0.00098886306265523775
0.001711725998522884	-0.0013901968381628592	0.00097884522055535945
0.00178858019030063	-0.0014060013597168744	0.00096551883122607523
0.0018584122758319708	-0.0014172056781641736	0.00094948306886115559
0.0019209020069706844	-0.0014234973190842882	0.00093029898870847097
0.0019762034734599168	-0.0014252804551111897	0.00090858280332597064
0.0020240509278105371	-0.0014222377203500633	0.00088392441013306023
0.0020646536032935953	-0.00141481951793788	0.00085694866961545831
0.0020977677228874157	-0.0014027317246174749	0.0008272641407302073
0.002123668644488738	-0.0013864526074882015	0.00079550675834895836
0.0021421442862359038	-0.0013657127342492389	0.00076129813782945584
0.0021535208769473055	-0.0013410173964932773	0.00072528071127283492
0.0021576220867702972	-0.0013121193595638679	0.00068708543961909507
0.0021548227178795369	-0.0012795418118353229	0.00064737426310801892
0.0021449802674732739	-0.0012430616324811466	0.00060578186324883754
0.0021285063696655062	-0.0012032278838453804	0.00056298040456686724
0.0021053012939125067	-0.0011598302779329116	0.00051860232221983106
0.0020758125538372502	-0.0011134401405020092	0.00047332593025963899
0.0020399698852100433	-0.0010638538838922081	0.00042679436899666599
0.0019982582438824851	-0.0010116627618070345	0.00037970072685910823
0.0019506292268018331	-0.00095668104585849064	0.00033167218432164462
0.0018976034746414901	-0.00089951831183766526	0.00028338725464581438
0.0018391632774067115	-0.0008399787659662452	0.00023451273096736528
0.0017758566019373228	-0.00077868551512995984	0.00018570946084584417
0.0017076733540648025	-0.00071547103181023594	0.00013662096835159558
0.0016351968150122133	-0.00065093968380916566	0.000087945383412639608
0.0015584389686721825	-0.00058493577498299181	0.000039289507771337641
0.0014779813065405115	-0.00051808376411069607	-0.0000086400182190129668
0.0013938701174600748	-0.00045020665635144095	-0.000056251952617956524

-0.00010283845934892658	0.00030743712827310333	-0.00033000377113186338
-0.00014882188419158707	0.00033256838667891598	-0.00033928679188414751
-0.0001934940361235705	0.00035595388932737866	-0.00034687885169742538
-0.00023728802129765201	0.00037827068111251621	-0.00035371384274510967
-0.00027950889528369736	0.00039872917268365905	-0.00035882381207530661
-0.00032060773623635583	0.00041801755730318853	-0.00036315422265930253
-0.00035989449581343407	0.00043534102299676476	-0.00036576311987010978
-0.00039783583101821347	0.0004514066315465747	-0.00036763905727959676
-0.00043374133174993142	0.00046544168582642476	-0.00036781358900952674
-0.00046810321671730522	0.00047818242697997542	-0.0003672050119213374
-0.00050025083686440789	0.00048882244495281708	-0.00036498634989638604
-0.00053069564455483862	0.00049810541327826316	-0.00036204393068751299
-0.00055874335165178365	0.00050531036238359852	-0.00035748270164706391
-0.00058494934563631728	0.00051112816544740926	-0.00035228747543676001
-0.0006086545449013347	0.0005148548291701088	-0.00034551404262245667
-0.00063038469798899115	0.0005172270946876942	-0.00033817688481614606
-0.00064952760312150457	0.00051751170545672343	-0.00032933008835131982
-0.00066662365005450011	0.00051648051692205913	-0.00032000760479768738
-0.00068105820536378198	0.00051339669143979231	-0.00030926083619526624
-0.00069340277944629491	0.00050905874348756725	-0.00029813612119934619
-0.00070304684983063742	0.00050272864842492316	-0.00028567932086350596
-0.00071059092190144807	0.0004952273344279879	-0.00027293984093171081
-0.00071543124093910926	0.0004858145033054726	-0.00025895826823021592
-0.00071819809114411848	0.0004753189165866735	-0.00024480524266682051
-0.00071829119544377078	0.0004630009194159148	-0.00022952522075358117
-0.0007163559127458057	0.00044971140783373661	-0.00021421312632923569
-0.00071180085541535168	0.00043471981145224011	-0.00019788626118805381
-0.00070529402306138654	0.00041890012353078786	-0.00018162289447892256
-0.00069625949971620013	0.00040150460778300969	-0.00016441022240285636
-0.00068538601691430162	0.00038340241243765968	-0.00014746101937459545
-0.00067209173575583005	0.00036385424336953442	-0.0001296961492806099
-0.00065707496656227608	0.00034380458315989024	-0.00011222576604668537
-0.00063978359363386006	0.00032242749378139047	-0.000094129058136657034
-0.00062095145155948101	0.00030068328278034899	-0.000076443984548822501
-0.0005999707441854598	0.00027781006367451778	-0.000058239219929641633
-0.00057763247123221841	0.00025471111246339138	-0.000040584325141533248
-0.00055333879128422348	0.00023065388526395799	-0.000022516456553598949
-0.00052786055412808348	0.00020654724271713465	-0.0000051183448352880648
-0.00050063490924587284	0.00018164959434224673	0.000012593323469278252
-0.00047243124842208687	0.00015687654067901352	0.000029520662130661391
-0.00044269464033252137	0.00013147613766815655	0.000046660621387630604
-0.00041219815242038698	0.0001063719928901365	0.000062899756543591603
-0.00038038916321840935	0.000080802586229005487	0.000079243662698242311
-0.00034804414435529512	0.000055706747009405706	0.000094580277856859841
-0.0003146096584042469	0.000030317385361566199	0.00010993965360518287
-0.0002808786205883095	0.0000055721892928241659	0.0001242191665805 45
-0.00024629707240219527	-0.000019313533840499119	0.00013845596299060653
-0.00021166052750568737	-0.000043412155480999221	0.00015151427514765309
-0.00017640582433084764	-0.000067509213900208195	0.00016439927787793591
-0.00014132351978431807	-0.000090653734473817704	0.0001760454230776724
-0.00010585526734065761	-0.00011362489670508789	0.0001875804189429048
-0.000070807489690653033	-0.00013553943008691319	0.00019769923111891781
-0.000035610778606344484	-0.00015717767311484608	0.00020766819328766652
-0.0000010267482754306584	-0.00017755838795338206	0.00021623251675856542
0.000033475843757728007	-0.00019758605232928547	0.0002245683064246687
0.000067118977783893041	-0.00021624181816448521	0.00023148137846121632
0.0001005048128976467 2	-0.00023442666264361558	0.00023813352544945323
0.00013280490625410543	-0.00025115486712243111	0.00024333464088812772
0.00016465250338366976	-0.00026732115050411747	0.00024825316677369938
0.00019522728378608759	-0.0002819486752807473	0.00025170386981805074
0.00022516829469897484	-0.00029593552328226755	0.00025487186623017596
0.00025366149574817089	-0.00030831295608087835	0.00025657491656392185
0.00028135355724374271	-0.0003199887223277 3734	0.00025801994090392157

0.00025801757303272428	-0.0001469677393142084	0.0001101646465786546
0.00025777535551880929	-0.00014263929775265756	0.0001071759235033139
0.0002560887810210662	-0.00013712775306687353	0.00010663680022482414
0.00025416987591782465	-0.00013196846615529329	0.00010565468409777165
0.00025084267103707078	-0.00012593738245925959	0.000099416577350158741
0.00024737284335290969	-0.00012003916396033289	0.00009905842015937253
0.00024260421142313247	-0.00011300444746429206	0.00009513236986817596
0.00023758886451544139	-0.00010670007151901287	0.000093974108573662354
0.00023133621975764953	-0.000099118811303544781	0.000090499805870497465
0.00022503037344462698	-0.000092260090378026936	0.000089325779675198443
0.00021742478629184362	-0.000084253300525510139	0.000086197584639391249
0.00020987034837344773	-0.000076993588255282186	0.000085195261840568907
0.00020108542395707446	-0.000068645140897852404	0.000082428641002914652
0.00019242422508625959	-0.000061092219468918303	0.000081709756873558307
0.000182610514700927	-0.00005249286518665356	0.000079345887525347249
0.00017300604705350374	-0.000044742098498718689	0.000078997189411837698
0.00016232659440101275	-0.000035981083891447208	0.000077085729392808045
0.00015193707916485479	-0.000028138140048195297	0.00007717836162892712
0.00014054096516596557	-0.000019329086894088066	0.00007577032092086558
0.00012951010341717174	-0.000011536458197057044	0.000076366567932588132
0.00011753731515115953	-0.0000028403160805324547	0.000075508158206514468
0.00010602380766471312	0.0000047232471735315223	0.000076661984535180382
0.000093660906896834723	0.000013169759800744398	0.000076395283552097536
0.000081881745329259042	0.000020447135008449992	-0.0028198205780140565
0.000069337865002288158	0.00002886904426262259	
0.000057445101527889113	0.000036210450885439723	
0.000044749488283311097	0.000043687605142000464	
0.000032867650288383263	0.000050582272923321934	
0.000020444325312044768	0.000057951318511700618	
0.0000086902876342089853	0.000064201533892440082	
-0.0000034803948071975011	0.00007110013761986618	
-0.000014834724958041542	0.000076767944085206	
-0.000026601762226956694	0.000083097395446525644	
-0.000037450431234486992	0.000088126871241741515	
-0.000048666047744277876	0.000093825457333526589	
-0.000058886689202743234	0.000098173655417675438	
-0.000069423455994695009	0.00010320746336116684	
-0.000078889124084297914	0.00010686291664000142	
-0.000088612753633763534	0.00011124376365828029	
-0.000097193718518075525	0.00011423751416412371	
-0.00010597953615275367	0.00011802771549413414	
-0.0001135737597760581	0.00012041879593506933	
-0.00012135786818006939	0.0001236783306207581	
-0.00012793971408940705	0.00012540150196365838	
-0.00013471346760123622	0.00012793574700008335	
-0.00014023369232932844	0.0001279264948065204	
-0.00014583198275159833	0.00013027801303914458	
-0.00015004481167425884	0.00013076832806300587	
-0.00015467113078508708	0.00013165286858759411	
-0.00015777182536080327	0.00013155945975404718	
-0.00016108318302178806	0.00013196984735976093	
-0.00016306276417511626	0.00013127836665178865	
-0.00016517719358254352	0.00013116893748943773	
-0.00016596415684399764	0.0001299515207842006	
-0.0001668956216989495	0.00012933937322024769	
-0.00016650515906696832	0.00012764690899380897	
-0.0001662812980484478	0.00012654926738739011	
-0.00016475565789722708	0.00012442771496268355	
-0.00016343457873385864	0.00012284850378707461	
-0.0001608411489000673	0.00012034812217346291	
-0.00015849098742972681	0.00011827917561999532	
-0.00015489303178660039	0.00011549092437890653	
-0.00015155154684283901	0.00011292724537424661	

2Mfs_16Klpf_fir45 Filter Coefficients

Generated by MATLAB(R) 7.8 and the Signal Processing Toolbox 6.11.
Generated on: 22-Jun-2009 15:59:36
Coefficient Format: Decimal
Discrete-Time FIR Filter (real)
Filter Structure : Direct-Form FIR
Filter Length : 1234
Stable : Yes
Linear Phase : Yes (Type 2)

Numerator:

```
-0.0028198205780140565
 0.000076395283552097536
 0.000076661984535180382
 0.000075508158206514468
 0.000076366567932588132
 0.00007577032092086558
 0.00007717836162892712
 0.000077085729392808045
 0.000078997189411837698
 0.000079345887525347249
 0.000081709756873558307
 0.000082428641002914652
 0.000085195261840568907
 0.000086197584639391249
 0.000089325779675198443
 0.000090499805870497465
 0.000093974108573662354
 0.00009513236986817596
 0.00009905842015937253
 0.000099416577350158741
 0.00010565468409777165
 0.00010663680022482414
 0.0001071759235033139
 0.0001101646465786546
 0.00011292724537424661
 0.00011549092437890653
 0.00011827917561999532
 0.00012034812217346291
 0.00012284850378707461
 0.00012442771496268355
 0.00012654926738739011
 0.00012764690899380897
 0.00012933937322024769
 0.0001299515207842006
 0.00013116893748943773
 0.00013127836665178865
 0.00013196984735976093
 0.00013155945975404718
 0.00013165286858759411
 0.00013076832806300587
 0.00013027801303914458
 0.0001279264948065204
 0.00012793574700008335
 0.00012540150196365838
 0.0001236783306207581
 0.00012041879593506933
 0.00011802771549413414
 0.00011423751416412371
 0.00011124376365828029
 0.00010686291664000142
 0.00010320746336116684
 0.000098173655417675438
 0.000093825457333526589
 0.000088126871241741515
 0.000083097395446525644
 0.000076767944085206
 0.00007110013761986618
 0.000064201533892440082
 0.000057951318511700618
 0.000050582272923321934
 0.000043687605142000464
 0.000036210450885439723
 0.00002886904426262259
 0.000020447135008449992
 0.000013169759800744398
 0.0000047232471735315223
-0.0000028403160805324547
-0.000011536458197057044
-0.000019329086894088066
-0.000028138140048195297
-0.000035981083891447208
-0.000044742098498718689
-0.00005249286518665356
-0.000061092219468918303
-0.000068645140897852404
-0.000076993588255282186
-0.000084253300525510139
-0.000092260090378026936
-0.000099118811303544781
-0.00010670007151901287
-0.00011300444746429206
-0.00012003916396033289
-0.00012593738245925959
-0.00013196846615529329
-0.00013712775306687353
-0.00014263929775265756
-0.0001469677393142084
-0.00015155154684283901
-0.00015489303178660039
-0.00015849098742972681
-0.0001608411489000673
-0.00016343457873385864
-0.00016475565789722708
-0.0001662812980484478
-0.00016650515906696832
-0.0001668956216989495
-0.00016596415684399764
-0.00016517719358254352
-0.00016306276417511626
   -0.00016108318302178806
   -0.00015777182536080327
-0.00015467113078508708
-0.00015004481167425884
-0.00014583198275159833
-0.00014023369232932844
-0.00013471346760123622
-0.00012793971408940705
-0.00012135786818006939
-0.0001135737597760581
-0.00010597953615275367
-0.000097193718518075525
-0.000088612753633763534
-0.000078889124084297914
-0.000069423455994695009
-0.000058886689202743234
-0.000048666047744277876
-0.000037450431234486992
-0.000026601762226956694
-0.000014834724958041542
-0.0000034803948071975011
 0.0000086902876342089853
 0.000020444325312044768
 0.000032867650288383263
 0.000044749488283311097
 0.000057445101527889113
 0.000069337865002288158
 0.000081881745329259042
 0.000093660906896834723
 0.00010602380766471312
 0.00011753731515115953
 0.00012951010341717174
 0.00014054096516596557
 0.00015193707916485479
 0.00016232659440101275
 0.00017300604705350374
 0.000182610514700927
 0.00019242422508625959
 0.00020108542395707446
 0.00020987034837344773
```

0.00021742478629184362	-0.00036498634989638604	0.00048882244495281708
0.00022503037344462698	-0.0003672050119213374	0.00047818242697997542
0.00023133621975764953	-0.00036781358900952674	0.00046544168582642476
0.00023758886451544139	-0.00036763905727959676	0.0004514066315465747
0.00024260421142313247	-0.00036576311987010978	0.00043534102299676476
0.00024737284335290969	-0.00036315422265930253	0.00041801755730318853
0.00025084267103707078	-0.00035882381207530661	0.00039872917268365905
0.00025416987591782465	-0.00035371384274510967	0.00037827068111251621
0.0002560887810210662	-0.00034687885169742538	0.00035595388932737866
0.00025777535551880929	-0.00033928679188414751	0.00033256838667891598
0.00025801757303272428	-0.00033000377113186338	0.00030743712827310333
0.00025801994090392157	-0.00031998872232773734	0.00028135355724374271
0.00025657491656392185	-0.00030831295608087835	0.00025366149574817089
0.00025487186623017596	-0.00029593552328226755	0.00022516829469897484
0.00025170386981805074	-0.0002819486752807473	0.00019522728378608759
0.00024825316677369938	-0.00026732115050411747	0.00016465250338366976
0.00024333464088812772	-0.00025115486712243111	0.00013280490625410543
0.00023813352544945323	-0.00023442666264361558	0.00010050481289764672
0.00023148137846121632	-0.00021624181816448521	0.000067118977783893041
0.0002245683064246687	-0.00019758605232928547	0.000033475843757728007
0.00021623251675856542	-0.00017755838795338206	-0.0000010267482754306584
0.00020766819328766652	-0.00015717767311484608	-0.000035610778606344484
0.00019769923111891781	-0.00013553943008691319	-0.000070807489690653033
0.0001875804189429048	-0.00011362489670508789	-0.00010585526734065761
0.0001760454230776724	-0.000090653734473817704	-0.00014132351978431807
0.00016439927787793591	-0.000067509213900208195	-0.00017640582433084764
0.00015151427514765309	-0.000043412155480999221	-0.00021166052750568737
0.00013845596299060653	-0.000019313533840499119	-0.00024629707240219527
0.00012421916658054 5	0.0000055721892928241659	-0.0002808786205883095
0.00010993965360518287	0.000030317385361566199	-0.0003146096584042469
0.000094580277856859841	0.000055706747009405706	-0.00034804414435529512
0.000079243662698242311	0.000080802586229005487	-0.00038038916321840935
0.000062899756543591603	0.0001063719928901365	-0.00041219815242038698
0.000046660621387630604	0.00013147613766815655	-0.00044269464033252137
0.000029520662130661391	0.00015687654067901352	-0.00047243124842208687
0.000012593323469278252	0.00018164959434224673	-0.00050063490924587284
-0.0000051183448352880648	0.00020654724271713465	-0.00052786055412808348
-0.000022516456553598949	0.00023065388526395799	-0.00055333879128422348
-0.000040584325141533248	0.00025471111246339138	-0.00057763247123221841
-0.000058239219929641633	0.00027781006367451778	-0.0005999707441854598
-0.000076443984548822501	0.00030068328278034899	-0.00062095145155948101
-0.000094129058136657034	0.00032242749378139047	-0.00063978359363386006
-0.00011222576604668537	0.00034380458315989024	-0.00065707496656227608
-0.0001296961492806099	0.00036385424336953442	-0.00067209173575583005
-0.00014746101937459545	0.00038340241243765968	-0.00068538601691430162
-0.00016441022240285636	0.00040150460778300969	-0.00069625949971620013
-0.00018162289447892256	0.00041890012353078786	-0.00070529402306138654
-0.00019788626118805381	0.00043471981145224011	-0.00071180085541535168
-0.00021421312632923569	0.00044971140783373661	-0.0007163559127458057
-0.00022952522075358117	0.0004630009194159148	-0.00071829119544377078
-0.00024480524266682051	0.0004753189165866735	-0.00071819809114411848
-0.00025895826823021592	0.0004858145033054726	-0.00071543124093910926
-0.00027293984093171081	0.0004952273344279879	-0.00071059092190144807
-0.00028567932086350596	0.00050272864842492316	-0.00070304684983063742
-0.00029813612119934619	0.00050905874348756725	-0.00069340277944629491
-0.00030926083619526624	0.00051339669143979231	-0.00068105820536378198
-0.00032000760479768738	0.00051648051692205913	-0.00066662365005450011
-0.00032933008835131982	0.00051751170545672343	-0.00064952760312150457
-0.00033817688481614606	0.0005172270946876942	-0.00063038469798899115
-0.00034551404262245667	0.0005148548291701088	-0.0006086545449013347
-0.00035228747543676001	0.00051112816544740926	-0.00058494934563631728
-0.00035748270164706391	0.00050531036238359852	-0.00055874335165178365
-0.00036204393068751299	0.00049810541327826316	-0.00053069564455483862

-0.00050025083686440789	0.00028566617290912231	0.00033102966487666028
-0.00046810321671730522	0.0002234915111891775	0.00043198239322995904
-0.00043374133174993142	0.00015956522944163189	0.00053319669279704076
-0.00039783583101821347	0.000094523149082608674	0.00063404262179773572
-0.00035989449581343407	0.00002812166612714594	0.00073449672052161401
-0.00032060773623635583	-0.000038999752333575277	0.00083392434356337718
-0.00027950889528369736	-0.0001070705633499933	0.00093229600508296486
-0.00023728802129765201	-0.00017544612161619986	0.0010289748704705509
-0.0001934940361235705	-0.00024434363293221722	0.0011239203104572819
-0.00014882188419158707	-0.00031310817917260718	0.0012165099072220134
-0.00010283845934892658	-0.00038194519676740127	0.0013066965951852518
-0.000056251952617956524	-0.00045020665635144095	0.0013938701174600748
-0.0000086400182190129668	-0.00051808376411069607	0.0014779813065405115
0.000039289507771337641	-0.00058493577498299181	0.0015584389686721825
0.000087945383412639608	-0.00065093968380916566	0.0016351968150122133
0.00013662096835159558	-0.00071547103181023594	0.0017076733540648025
0.00018570946084584417	-0.00077868551512995984	0.0017758566019373228
0.00023451273096736528	-0.0008399787659662452	0.0018391632774067115
0.00028338725464581438	-0.00089951831183766526	0.0018976034746414901
0.00033167218432164462	-0.00095668104585849064	0.0019506292268018331
0.00037970072685910823	-0.0010116627618070345	0.0019982582438824851
0.00042679436899666599	-0.0010638538838922081	0.0020399698852100433
0.00047332593025963899	-0.0011134401405020092	0.0020758125538372502
0.00051860232221983106	-0.0011598302779329116	0.0021053012939125067
0.00056298040456686724	-0.0012032278838453804	0.0021285063696655062
0.00060578186324883754	-0.0012430616324811466	0.0021449802674732739
0.00064737426310801892	-0.0012795418118353229	0.0021548227178795369
0.00068708543961909507	-0.0013121193595638679	0.0021576220867702972
0.00072528071127283492	-0.0013410173964932773	0.0021535208769473055
0.00076129813782945584	-0.0013657127342492389	0.0021421442862359038
0.00079550675834895836	-0.0013864526074882015	0.002123668644488738
0.0008272641407302073	-0.0014027317246174749	0.0020977677228874157
0.00085694866961545831	-0.00141481951793788	0.0020646536032935953
0.00088392441013306023	-0.0014222377203500633	0.0020240509278105371
0.00090858280332597064	-0.0014252804551111897	0.0019762034734599168
0.00093029898870847097	-0.0014234973190842882	0.0019209020069706844
0.00094948306886115559	-0.0014172056781641736	0.0018584122758319708
0.00096551883122607523	-0.0014060013597168744	0.00178858019030063
0.00097884522055535945	-0.0013901968381628592	0.001711725998522884
0.00098886306265523775	-0.0013694453086006233	0.0016277302118003771
0.00099600124396552643	-0.0013440836910837044	0.0015369611061019496
0.00099972276851222151	-0.0013137815727844219	0.0014393570131921623
0.0010004436427659305	-0.0012789180941528548	0.0013353241058304483
0.0009976418991891659	-0.00123920249922804	0.0012248460292908921
0.00099177238342760187	-0.0011950339802983619	0.001108373499420467
0.00098233704063171697	-0.001146151770691882	0.00098594210839710429
0.00096979732956396563	-0.0010929871045531774	0.00085804132602843074
0.00095367757260468762	-0.0010353146297528912	0.00072476013817669195
0.00093446318622413807	-0.00097359278060084927	0.0005866221539868547
0.00091170537202771599	-0.00090763363368269109	0.00044375834241092843
0.00088590843103020494	-0.00083791359528353343	0.00029673866896819806
0.00085664890063087392	-0.00076428088185246084	0.00014573082758903653
0.00082444120729077046	-0.00068724218955202665	-0.0000086564015360848789
0.00078889517163202409	-0.00060667297521134825	-0.00016622086878493409
0.00075054317511755349	-0.00052310764577098495	-0.00032631461012709324
0.00070902141967763834	-0.00043644565550454836	-0.00048870637059933355
0.00066487861517135682	-0.00034725380307260544	-0.00065272219431096803
0.00061777520096100661	-0.00025544142663221298	-0.00081808161842529883
0.00056828027381864528	-0.00016161206312381715	-0.00098410743222504624
0.00051605786594639044	-0.000065702752502615723	-0.0011504811011481294
0.00046172646690290781	0.00003168649933427874	-0.0013165020184760152
0.0004049518838699479	0.00013057484784520989	-0.0014818362815834157
0.0003463592226354709	0.00023035256527318988	-0.0016457696457433247

-0.0018079424358925069	0.0075982462309454098	0.011953183474458288
-0.0019676314824506828	0.008046398944676925	0.011539889012966651
-0.0021244684162301363	0.008493889137702328	0.011119581573262106
-0.0022777242431054898	0.0089398689411820022	0.010692987558016571
-0.0024270242375341805	0.0093835861827764409	0.010260944031035897
-0.0025716450132131493	0.0098241950111563683	0.0098241950111563683
-0.0027111985825774912	0.010260944031035897	0.0093835861827764409
-0.0028449779801232996	0.010692987558016571	0.0089398689411820022
-0.0029725973714984756	0.011119581573262106	0.008493889137702328
-0.0030933621018964901	0.011539889012966651	0.008046398944676925
-0.0032068960700438291	0.011953183474458288	0.0075982462309454098
-0.0033125237060962903	0.012358636335892102	0.0071501596285552639
-0.003409889186666867	0.01275554538533619	0.0067029885555287194
-0.00349832747214672	0.013143114155810359	0.0062574391747704456
-0.0035775230908427959	0.013520652773112391	0.0058143447807188931
-0.0036468327778465997	0.013887407951264445	0.0053743900665045334
-0.0037059621870312613	0.014242718936035882	0.0049383867998628334
-0.0037543160508810725	0.014585868949721085	0.0045069811540848206
-0.0037916266700765309	0.014916236624631695	0.0040809603341974724
-0.003817335916435763	0.015233156610208881	0.0036609372966672667
-0.0038312189241855418	0.015536047481389542	0.0032476583127148788
-0.0038327668193246542	0.015824294243633744	0.0028416928943233071
-0.0038217931023824519	0.016097371092578854	0.0024437591969583847
-0.0037978395880742455	0.016354714760492963	0.0020543597625958679
-0.0037607694175273224	0.016595854196395494	0.0016741798841105676
-0.0037101727706956925	0.016820299830547555	0.0013036664877290158
-0.003645971616153181	0.017027627050223024	0.00094345352587472476
-0.0035678169838762605	0.017217420639752169	0.00059393104152329155
-0.00347567709518157	0.017389315574409342	0.0002556790912609506
-0.0033692729684057475	0.017542972127936346	-0.000070970359776215391
-0.003248624726074058	0.017678087036844384	-0.00038550428915550125
-0.0031135249699860701	0.017794398078673217	-0.00068764466145177774
-0.0029640445966621761	0.017891678630440701	-0.00097693916461130074
-0.0028000546804664754	0.017969729342138967	-0.0012531809713051878
-0.0026216838341772009	0.018028406005298959	-0.0015159742108734864
-0.0024288628261028575	0.018067586331780183	-0.0017651855228356887
-0.0022217988138884632	0.018087195053390105	-0.0020004832370599376
-0.0020004832370599376	0.018087195053390105	-0.0022217988138884632
-0.0017651855228356887	0.018067586331780183	-0.0024288628261028575
-0.0015159742108734864	0.018028406005298959	-0.0026216838341772009
-0.0012531809713051878	0.017969729342138967	-0.0028000546804664754
-0.00097693916461130074	0.017891678630440701	-0.0029640445966621761
-0.00068764466145177774	0.017794398078673217	-0.0031135249699860701
-0.00038550428915550125	0.017678087036844384	-0.003248624726074058
-0.000070970359776215391	0.017542972127936346	-0.0033692729684057475
0.0002556790912609506	0.017389315574409342	-0.00347567709518157
0.00059393104152329155	0.017217420639752169	-0.0035678169838762605
0.00094345352587472476	0.017027627050223024	-0.003645971616153181
0.0013036664877290158	0.016820299830547555	-0.0037101727706956925
0.0016741798841105676	0.016595854196395494	-0.0037607694175273224
0.0020543597625958679	0.016354714760492963	-0.0037978395880742455
0.0024437591969583847	0.016097371092578854	-0.0038217931023824519
0.0028416928943233071	0.015824294243633744	-0.0038327668193246542
0.0032476583127148788	0.015536047481389542	-0.0038312189241855418
0.0036609372966672667	0.015233156610208881	-0.003817335916435763
0.0040809603341974724	0.014916236624631695	-0.0037916266700765309
0.0045069811540848206	0.014585868949721085	-0.0037543160508810725
0.0049383867998628334	0.014242718936035882	-0.0037059621870312613
0.0053743900665045334	0.013887407951264445	-0.0036468327778465997
0.0058143447807188931	0.013520652773112391	-0.0035775230908427959
0.0062574391747704456	0.013143114155810359	-0.00349832747214672
0.0067029885555287194	0.01275554538533619	-0.003409889186666867
0.0071501596285552639	0.012358636335892102	-0.0033125237060962903

-0.0032068960700438291	0.0013066965951852518	-0.00038194519676740127
-0.0030933621018964901	0.0012165099072220134	-0.00031310817917260718
-0.0029725973714984756	0.0011239203104572819	-0.00024434363293221722
-0.0028449779801232996	0.0010289748704705509	-0.00017544612161619986
-0.0027111985825774912	0.00093229600508296486	-0.0001070705633499933
-0.0025716450132131493	0.00083392434356337718	-0.000038999752333575277
-0.0024270242375341805	0.00073449672052161401	0.0000281216661271459 4
-0.0022777242431054898	0.00063404262179773572	0.000094523149082608674
-0.0021244684162301363	0.00053319669279704076	0.00015956522944163189
-0.0019676314824506828	0.00043198239322995904	0.0002234915111891775
-0.0018079424358925069	0.00033102966487666028	0.00028566617290912231
-0.0016457696457433247	0.00023035256527318988	0.0003463592226354709
-0.0014818362815834157	0.00013057484784520989	0.0004049518838699479
-0.0013165020184760152	0.000031686499334278 74	0.00046172646690290781
-0.0011504811011481294	-0.000065702752502615723	0.00051605786594639044
-0.00098410743222504624	-0.00016161206312381715	0.00056828027381864528
-0.00081808161842529883	-0.00025544142663221298	0.00061777520096100661
-0.00065272219431096803	-0.00034725380307260544	0.00066487861517135682
-0.00048870637059933355	-0.00043644565550454836	0.00070902141967763834
-0.00032631461012709324	-0.00052310764577098495	0.00075054317511755349
-0.00016622086878493409	-0.00060667297521134825	0.00078889517163202409
-0.0000086564015360848789	-0.00068724218955202665	0.00082444120729077046
0.00014573082758903653	-0.00076428088185246084	0.00085664890063087392
0.00029673866896819806	-0.00083791359528353343	0.00088590843103020494
0.00044375834241092843	-0.00090763363368269109	0.00091170537202771599
0.0005866221539868547	-0.00097359278060084927	0.00093446318622413807
0.00072476013817669195	-0.0010353146297528912	0.00095367757260468762
0.00085804132602843074	-0.0010929871045531774	0.00096979732956396563
0.00098594210839710429	-0.001146151770691882	0.00098233704063171697
0.001108373499420467	-0.0011950339802983619	0.00099177238342760187
0.0012248460292908921	-0.00123920249922804	0.0009976418991891659
0.0013353241058304483	-0.0012789180941528548	0.0010004436427659305
0.0014393570131921623	-0.0013137815727844219	0.00099972276851222151
0.0015369611061019496	-0.0013440836910837044	0.00099600124396552643
0.0016277302118003771	-0.0013694453086006233	0.00098886306265523775
0.001711725998522884	-0.0013901968381628592	0.00097884522055535945
0.00178858019030063	-0.0014060013597168744	0.00096551883122607523
0.0018584122758319708	-0.0014172056781641736	0.00094948306886115559
0.0019209020069706844	-0.0014234973190842882	0.00093029898870847097
0.0019762034734599168	-0.0014252804551111897	0.00090858280332597064
0.0020240509278105371	-0.0014222377203500633	0.00088392441013306023
0.0020646536032935953	-0.00141481951793788	0.00085694866961545831
0.0020977677228874157	-0.0014027317246174749	0.0008272641407302073
0.002123668644488738	-0.0013864526074882015	0.00079550675834895836
0.0021421442862359038	-0.0013657127342492389	0.00076129813782945584
0.0021535208769473055	-0.0013410173964932773	0.00072528071127283492
0.0021576220867702972	-0.0013121193595638679	0.00068708543961909507
0.0021548227178795369	-0.0012795418118353229	0.00064737426310801892
0.0021449802674732739	-0.0012430616324811466	0.00060578186324883754
0.0021285063696655062	-0.0012032278838453804	0.00056298040456686724
0.0021053012939125067	-0.0011598302779329116	0.00051860232221983106
0.0020758125538372502	-0.0011134401405020092	0.00047332593025963899
0.0020399698852100433	-0.0010638538838922081	0.00042679436899666599
0.0019982582438824851	-0.0010116627618070345	0.00037970072685910823
0.0019506292268018331	-0.00095668104585849064	0.00033167218432164462
0.0018976034746414901	-0.00089951831183766526	0.00028338725464581438
0.0018391632774067115	-0.0008399787659662452	0.00023451273096736528
0.0017758566019373228	-0.00077868551512995984	0.00018570946084584417
0.0017076733540648025	-0.00071547103181023594	0.00013662096835159558
0.0016351968150122133	-0.00065093968380916566	0.000087945383412639608
0.0015584389686721825	-0.00058493577498299181	0.000039289507771337641
0.0014779813065405115	-0.00051808376411069607	-0.0000086400182190129668
0.0013938701174600748	-0.00045020665635144095	-0.000056251952617956524

-0.00010283845934892658	0.00030743712827310333	-0.00033000377113186338
-0.00014882188419158707	0.00033256838667891598	-0.00033928679188414751
-0.0001934940361235705	0.00035595388932737866	-0.00034687885169742538
-0.00023728802129765201	0.00037827068111251621	-0.00035371384274510967
-0.00027950889528369736	0.00039872917268365905	-0.00035882381207530661
-0.00032060773623635583	0.00041801755730318853	-0.00036315422265930253
-0.00035989449581343407	0.00043534102299676476	-0.00036576311987010978
-0.00039783583101821347	0.0004514066315465747	-0.00036763905727959676
-0.00043374133174993142	0.00046544168582642476	-0.00036781358900952674
-0.00046810321671730522	0.00047818242697997542	-0.0003672050119213374
-0.00050025083686440789	0.00048882244495281708	-0.00036498634989638604
-0.00053069564455483862	0.00049810541327826316	-0.00036204393068751299
-0.00055874335165178365	0.00050531036238359852	-0.00035748270164706391
-0.00058494934563631728	0.00051112816544740926	-0.00035228747543676001
-0.0006086545449013347	0.0005148548291701088	-0.00034551404262245667
-0.00063038469798899115	0.0005172270946876942	-0.00033817688481614606
-0.00064952760312150457	0.00051751170545672343	-0.00032933008835131982
-0.00066662365005450011	0.00051648051692205913	-0.00032000760479768738
-0.00068105820536378198	0.00051339669143979231	-0.00030926083619526624
-0.00069340277944629491	0.00050905874348756725	-0.00029813612119934619
-0.00070304684983063742	0.00050272864842492316	-0.00028567932086350596
-0.00071059092190144807	0.0004952273344279879	-0.00027293984093171081
-0.00071543124093910926	0.0004858145033054726	-0.00025895826823021592
-0.00071819809114411848	0.0004753189165866735	-0.00024480524266682051
-0.00071829119544377078	0.0004630009194159148	-0.00022952522075358117
-0.0007163559127458057	0.00044971140783373661	-0.00021421312632923569
-0.00071180085541535168	0.00043471981145224011	-0.00019788626118805381
-0.00070529402306138654	0.00041890012353078786	-0.00018162289447892256
-0.00069625949971620013	0.00040150460778300969	-0.00016441022240285636
-0.00068538601691430162	0.00038340241243765968	-0.00014746101937459545
-0.00067209173575583005	0.00036385424336953442	-0.0001296961492806099
-0.00065707496656227608	0.00034380458315989024	-0.00011222576604668537
-0.00063978359363386006	0.00032242749378139047	-0.000094129058136657034
-0.00062095145155948101	0.00030068328278034899	-0.000076443984548822501
-0.0005999707441854598	0.00027781006367451778	-0.000058239219929641633
-0.00057763247123221841	0.00025471111246339138	-0.000040584325141533248
-0.00055333879128422348	0.00023065388526395799	-0.000022516456553598949
-0.00052786055412808348	0.00020654724271713465	-0.0000051183448352880648
-0.00050063490924587284	0.00018164959434224673	0.000012593323469278252
-0.00047243124842208687	0.00015687654067901352	0.000029520662130661391
-0.00044269464033252137	0.00013147613766815655	0.000046660621387630604
-0.00041219815242038698	0.0001063719928901365	0.000062899756543591603
-0.00038038916321840935	0.000080802586229005487	0.000079243662698242311
-0.00034804414435529512	0.000055706747009405706	0.000094580277856859841
-0.0003146096584042469	0.000030317385361566199	0.00010993965360518287
-0.0002808786205883095	0.0000055721892928241659	0.000124219166580545
-0.00024629707240219527	-0.000019313533840499119	0.00013845596299060653
-0.00021166052750568737	-0.000043412155480999221	0.00015151427514765309
-0.00017640582433084764	-0.000067509213900208195	0.00016439927787793591
-0.00014132351978431807	-0.000090653734473817704	0.0001760454230776724
-0.00010585526734065761	-0.00011362489670508789	0.0001875804189429048
-0.000070807489690653033	-0.00013553943008691319	0.00019769923111891781
-0.000035610778606344484	-0.00015717767311484608	0.00020766819328766652
-0.0000010267482754306584	-0.00017755838795338206	0.00021623251675856542
0.000033475843757728007	-0.00019758605232928547	0.0002245683064246687
0.000067118977783893041	-0.00021624181816448521	0.00023148137846121632
0.00010050481289764672	-0.00023442666264361558	0.00023813352544945323
0.00013280490625410543	-0.00025115486712243111	0.00024333464088812772
0.00016465250338366976	-0.00026732115050411747	0.00024825316677369938
0.00019522728378608759	-0.0002819486752807473	0.00025170386981805074
0.00022516829469897484	-0.00029593552328226755	0.00025487186623017596
0.00025366149574817089	-0.00030831295608087835	0.00025657491656392185
0.00028135355724374271	-0.00031998872232773734	0.00025801994090392157

0.00025801757303272428
0.00025777535551880929
0.0002560887810210662
0.00025416987591782465
0.00025084267103707078
0.00024737284335290969
0.00024260421142313247
0.00023758886451544139
0.00023133621975764953
0.00022503037344462698
0.00021742478629184362
0.00020987034837344773
0.00020108542395707446
0.00019242422508625959
0.000182610514700927
0.00017300604705350374
0.00016232659440101275
0.00015193707916485479
0.00014054096516596557
0.00012951010341717174
0.00011753731515115953
0.00010602380766471312
0.000093660906896834723
0.000081881745329259042
0.000069337865002288158
0.000057445101527889113
0.000044749488283311097
0.000032867650288383263
0.000020444325312044768
0.0000086902876342089853
-0.0000034803948071975011
-0.000014834724958041542
-0.000026601762226956694
-0.000037450431234486992
-0.000048666047744277876
-0.000058886689202743234
-0.000069423455994695009
-0.000078889124084297914
-0.000088612753633763534
-0.000097193718518075525
-0.00010597953615275367
-0.0001135737597760581
-0.00012135786818006939
-0.00012793971408940705
-0.00013471346760123622
-0.00014023369232932844
-0.00014583198275159833
-0.00015004481167425884
-0.00015467113078508708
-0.00015777182536080327
-0.00016108318302178806
-0.00016306276417511626
-0.00016517719358254352
-0.00016596415684399764
-0.0001668956216989495
-0.00016650515906696832
-0.0001662812980484478
-0.00016475565789722708
-0.00016343457873385864
-0.0001608411489000673
-0.00015849098742972681
-0.00015489303178660039
-0.00015155154684283901
-0.0001469677393142084
-0.00014263929775265756
-0.00013712775306687353
-0.00013196846615529329
-0.00012593738245925959
-0.00012003916396033289
-0.00011300444746429206
-0.00010670007151901287
-0.000099118811303544781
-0.000092260090378026936
-0.000084253300525510139
-0.000076993588255282186
-0.000068645140897852404
-0.000061092219468918303
-0.00005249286518665356
-0.000044742098498718689
-0.000035981083891447208
-0.000028138140048195297
-0.000019329086894088066
-0.000011536458197057044
-0.0000028403160805324547
0.0000047232471735315223
0.000013169759800744398
0.000020447135008449992
0.00002886904426262259
0.000036210450885439723
0.000043687605142000464
0.000050582272923321934
0.000057951318511700618
0.000064201533892440082
0.00007110013761986618
0.000076767944085206
0.000083097395446525644
0.000088126871241741515
0.000093825457333526589
0.000098173655417675438
0.00010320746336116684
0.00010686291664000142
0.00011124376365828029
0.00011423751416412371
0.00011802771549413414
0.00012041879593506933
0.0001236783306207581
0.00012540150196365838
0.00012793574700008335
0.0001279264948065204
0.00013027801303914458
0.00013076832806300587
0.00013165286858759411
0.00013155945975404718
0.00013196984735976093
0.00013127836665178865
0.00013116893748943773
0.0001299515207842006
0.00012933937322024769
0.00012764690899380897
0.00012654926738739011
0.00012442771496268355
0.00012284850378707461
0.00012034812217346291
0.00011827917561999532
0.00011549092437890653
0.00011292724537424661
0.0001101646465786546
0.0001071759235033139
0.00010663680022482414
0.00010565468409777165
0.000099416577350158741
0.00009905842015937253
0.00009513236986817596
0.000093974108573662354
0.000090499805870497465
0.000089325779675198443
0.000086197584639391249
0.000085195261840568907
0.000082428641002914652
0.000081709756873558307
0.000079345887525347249
0.000078997189411837698
0.000077085729392808045
0.00007717836162892712
0.00007577032092086558
0.000076366567932588132
0.000075508158206514468
0.000076661984535180382
0.000076395283552097536
-0.0028198205780140565

40Kfs_1Klpf_fir45 Filter Coefficients

Generated by MATLAB(R) 7.8 and the Signal Processing Toolbox 6.11.
Generated on: 22-Jun-2009 16:00:29
Coefficient Format: Decimal
Discrete-Time FIR Filter (real)
Filter Structure : Direct-Form FIR
Filter Length : 67
Stable : Yes
Linear Phase : Yes (Type 1)

Numerator:

```
-0.0025692020393363129
 0.0013108283807641914
 0.0015771577302514694
 0.0020512449689182302
 0.0026049807098706768
 0.0031131997606313575
 0.0034517844562897836
 0.0035034029752609936
 0.003166396489479569
 0.0023660882460249922
 0.0010674714492398981
-0.00071343924506314779
-0.0029016880212675467
-0.005359878352184831
-0.0078902275029320932
-0.010243060225157702
-0.012131248720220471
-0.013249696937610614
-0.013299663883722026
-0.012016576420764825
-0.0091970018056605814
-0.004718549338757299
 0.0014443279951799606
 0.0091850362754643602
 0.018293976537851343
 0.028442838101723423
 0.039206423342546648
 0.050085008600304279
 0.060537909454668332
 0.070018239200955096
 0.078010693669522777
 0.084069147450203399
 0.087849408146234167
 0.089134251808535583
 0.087849408146234167
 0.084069147450203399
 0.078010693669522777
 0.070018239200955096
 0.060537909454668332
 0.050085008600304279
 0.039206423342546648
 0.028442838101723423
 0.018293976537851343
 0.0091850362754643602
 0.0014443279951799606
-0.004718549338757299
-0.0091970018056605814
-0.012016576420764825
-0.013299663883722026
-0.013249696937610614
-0.012131248720220471
-0.010243060225157702
-0.0078902275029320932
-0.005359878352184831
-0.0029016880212675467
-0.00071343924506314779
 0.0010674714492398981
 0.0023660882460249922
 0.003166396489479569
 0.0035034029752609936
 0.0034517844562897836
 0.0031131997606313575
 0.0026049807098706768
 0.0020512449689182302
 0.0015771577302514694
 0.0013108283807641914
-0.0025692020393363129
```

100Mfs 5Mlpf fir45 Filter Coefficients

Generated by MATLAB(R) 7.8 and the Signal Processing Toolbox 6.11.
Generated on: 22-Jun-2009 15:57:38
Coefficient Format: Decimal
Discrete-Time FIR Filter (real)
Filter Structure : Direct-Form FIR
Filter Length : 227
Stable : Yes
Linear Phase : Yes (Type 1)

Numerator:

```
 0.0029268708355747776
 0.00014835867564905303
 0.000055292110589805283
-0.000094222889210767322
-0.00028460411238011339
-0.00049238782334071879
-0.00069116232624750204
-0.00085216759479783862
-0.00095006806198992219
-0.00096439220775691313
-0.00088502706700675066
-0.00071143491122279258
-0.00045694266315562547
-0.0001447239058951118
 0.00019119370140131264
 0.00051231944042934513
 0.00077754553577085513
 0.0009509408514842871
 0.0010038506211300716
 0.00092199862336104783
 0.00070554160110106439
 0.00037278381077554497
-0.000043327977233012577
-0.00049565982422429425
-0.0009305504959257971
-0.0012909981394216327
-0.0015267905571478219
-0.0015980965324245062
-0.0014844956900632079
-0.0011855889892925838
-0.000726301328422472
-0.00015091628109066343
 0.00047604629681355972
 0.0010820990071740745
 0.0015901930737022994
 0.0019326848527425739
 0.0020549000570720482
 0.0019285748362172863
 0.0015492278836981086
 0.00095025766484596185
 0.00019588087317389929
-0.00064011576361726319
-0.0014521225694516455
-0.0021416299641440701
-0.002614135189679229
-0.0027980896300399813
-0.0026516091891805394
-0.0021719937423230842
-0.0013965451172977307
-0.00040204103698054765
 0.00070317687330379464
 0.001790382983512523
 0.0027246935269318083
 0.0033818439954535658
 0.0036635777450073309
 0.0035118181728722169
 0.0029178058135900706
 0.0019268730716514471
 0.00063541109722223526
-0.00081686849297404115
-0.0022621454309389743
-0.0035225848257078598
-0.0044319429266902908
-0.0048559889285257913
-0.0047117193922324588
-0.0039797127982873779
-0.0027116946232797548
-0.0010277516677746103
 0.0008930108007310744
 0.0028315425174784627
 0.0045511239541490417
 0.0058277541852092066
 0.0064740594075542836
 0.0063683650391263254
 0.0054706581600015682
 0.0038356863728265089
 0.0016103194963271342
-0.00097460592865133023
-0.0036315662890512644
-0.006040137059091448
-0.007886458032337805
-0.0089033883708013305
-0.0089002697638708508
-0.007796385255414812
-0.0056359071257421527
-0.0025939329288712258
 0.0010345367655420375
 0.0048588892042259494
 0.008431215210755 32
 0.011293026585278632
 0.013027463280167234
 0.01331119887609292
 0.011959192711049237
 0.0089592638587234867
 0.0044893363408995924
-0.0010830013493875483
-0.0072219828405844485
-0.013262922529682173
-0.018468920915524628
-0.022097225694199949
-0.023472022533763418
-0.022054449135794147
-0.017504027624480212
-0.0097244924717944208
 0.0011107042112556691
 0.014557344967922643
 0.029923784723321165
 0.046317758852374956
 0.06271396255888001
 0.07803433847009543
 0.091235529028716608
 0.10139870057252499
 0.10780271838726925
 0.10998997002182025
 0.10780271838726925
 0.10139870057252499
 0.091235529028716608
 0.07803433847009543
 0.06271396255888001
 0.046317758852374956
 0.029923784723321165
 0.014557344967922643
 0.0011107042112556691
-0.0097244924717944208
-0.017504027624480212
-0.022054449135794147
-0.023472022533763418
-0.022097225694199949
-0.018468920915524628
-0.013262922529682173
-0.0072219828405844485
-0.0010830013493875483
 0.0044893363408995924
 0.0089592638587234867
 0.011959192711049237
 0.01331119887609292
 0.013027463280167234
 0.011293026585278632
 0.00843121521075532
 0.0048588892042259494
 0.0010345367655420375
-0.0025939329288712258
-0.0056359071257421527
```

-0.007796385255414812	0.0036635777450073309	-0.0012909981394216327
-0.0089002697638708508	0.0033818439954535658	-0.0009305504959257971
-0.0089033883708013305	0.0027246935269318083	-0.00049565982422429425
-0.007886458032337805	0.001790382983512523	-0.000043327977233012577
-0.006040137059091448	0.00070317687330379464	0.00037278381077554497
-0.0036315662890512644	-0.00040204103698054765	0.00070554160110106439
-0.00097460592865133023	-0.0013965451172977307	0.00092199862336104783
0.0016103194963271342	-0.0021719937423230842	0.0010038506211300716
0.0038356863728265089	-0.0026516091891805394	0.0009509408514842871
0.0054706581600015682	-0.0027980896300399813	0.00077754553577085513
0.0063683650391263254	-0.002614135189679229	0.00051231944042934513
0.0064740594075542836	-0.0021416299641440701	0.00019119370140131264
0.0058277541852092066	-0.0014521225694516455	-0.0001447239058951118
0.0045511239541490417	-0.00064011576361726319	-0.00045694266315562547
0.0028315425174784627	0.00019588087317389929	-0.00071143491122279258
0.0008930108007310744	0.00095025766484596185	-0.00088502706700675066
-0.0010277516677746103	0.0015492278836981086	-0.00096439220775691313
-0.0027116946232797548	0.0019285748362172863	-0.00095006806198992219
-0.0039797127982873779	0.0020549000570720482	-0.00085216759479783862
-0.0047117193922324588	0.0019326848527425739	-0.00069116232624750204
-0.0048559889285257913	0.0015901930737022994	-0.00049238782334071879
-0.0044319429266902908	0.0010820990071740745	-0.00028460411238011339
-0.0035225848257078598	0.00047604629681355972	-0.000094222889210767322
-0.0022621454309389743	-0.00015091628109066343	0.000055292110589805283
-0.00081686849297404115	-0.000726301328422472	0.00014835867564905303
0.00063541109722223526	-0.0011855889892925838	0.0029268708355747776
0.0019268730716514471	-0.0014844956900632079	
0.0029178058135900706	-0.0015980965324245062	
0.0035118181728722169	-0.0015267905571478219	

Appendix E